SHARP

SHARP
COMPUTER
SOFTWARE

X68000用

C COMPILER PRO-68K ver2.1

拡張マニュアル

CZ-285LSD

はじめに

このたびは、「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ ver2.1」をお買い上げいただき、まことにありがとうございました。
「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ ver2.1」は、Ｘ６８０００のために作られた総合開発ツールです。ご使用に際しては、必ず本説明書に記載されている操作方法・注意事項をよくお読みいただき、正しい操作によって有効に活用されるようお願い致します。

※万一、ご不審な点・お気付きの点がありましたら、お買い求めの販売店、もしくは、もよりのシャープお客様相談窓口にご連絡ください。

商品構成は、下記の通りです。

ＸＣ　システムディスク１(3.5"FD,5"FD)・・・・・・・・・・・	各１枚
ＸＣ　システムディスク２(3.5"FD,5"FD)・・・・・・・・・・・	各１枚
ＸＣ　ライブラリディスク(3.5"FD,5"FD)・・・・・・・・・・・	各１枚
Ｃユーザーズマニュアル(ver2.0)・・・・・・・・・・・・・・	１冊
Ｃリファレンスマニュアル(ver2.0)・・・・・・・・・・・・・	１冊
ＣライブラリマニュアルＶＯＬ.1(ver2.0)・・・・・・・・・・	１冊
ＣライブラリマニュアルＶＯＬ.2(ver2.0)・・・・・・・・・・	１冊
ソースコードデバッガマニュアル(ver2.0)・・・・・・・・・・	１冊
アセンブラマニュアル(ver2.0)・・・・・・・・・・・・・・・	１冊
プログラマーズマニュアル(ver2.0)・・・・・・・・・・・・・	１冊
拡張マニュアル・・・・・・・・・・・・・・・・・・・・・・	１冊
登録カード・・・・・・・・・・・・・・・・・・・・・・・・	１枚

本マニュアルは「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ　ｖｅｒ２．１」で追加・変更された機能についてまとめてあります。まず最初に本書をお読みになってから、「Ｃユーザーズマニュアル」をご覧ください。

＜ご注意＞

１．本書の内容については万全を期して作成いたしましたが、万一ご不審な店や誤り記載もれなど、お気付きのことがありましたら、最寄りのシャープお客様ご相談窓口あるいはお買い求め販売店にご連絡ください。
２．運用した結果の影響については、１項にかかわらず責任をおいかねますのでご了承ください。
３．本書の内容に関しては、将来の予告なしに変更することがあります。
４．本書の内容の一部または全部を無断転載することは禁じられています。
５．付属の登録カードは必ず弊社までご返送ください。無登録の方は、一切のユーザーサポートが受けられませんので、ご注意ください。
６．本ソフトウェアを用いてソフト開発を行った商品を販売する場合の使用料（ロイアリティ）は無償ですが、マニュアルなどに本ソフトウェアを使用したことを明記ください。

ご使用になる前に（インストール）

「Ｃ compiler ＰＲＯ－６８Ｋ ｖｅｒ２．１」では、次のような追加／変更がされています。

１．ＦＭ音源／ＭＩＤＩ／ＡＤＰＣＭを同時に制御できるＯＰＭＤＲＶ３．Ｘを付属しています。
２．直接ＳＣＳＩを制御できるＳＣＳＩライブラリをサポートしています。
３．付属の浮動小数点・演算ドライバ（FLOAT2.X）に対応した高速ライブラリをサポートしています。
　　ＣＣに／Ｎｆスイッチを指定することによりＦＬＯＡＴドライバに依存しない高速な実行ファイルが作成できます。
　　（ＣＣのスイッチについては、「Ｃユーザーズマニュアル」を参照してください）

また、「Ｃユーザーズマニュアル」のオートインストールが次のように変更されています。

● オートインストール

ＯＰＭＤＲＶ３．Ｘの追加にともない「Ｃユーザーズマニュアル」のオートインストールの手順（１．４）に”ＭＵＳＩＣドライバ”の選択項目が追加されています。

例）フロピーディスクの場合

グラフィックＲＡＭ	RAM_DISKとして使用
Ｂａｓｉｃ	使用する
ＢＡＳＴＯＣ	使用する
ＭＵＳＩＣドライバ	ＯＰＭＤＲＶ３

＊なお、ｖｅｒ２．１ではビジュアルシェルは付属していません

標準では、今回追加されたＯＰＭＤＲＶ３．Ｘの環境がインストールされるよされるように設定されていますが、”する／しない”で”しない”を選択することによってＯＰＭＤＲＶ２．Ｘの環境がインストールされます。

ＯＰＭＤＲＶ２．Ｘ用のＸ－ＢＡＳＩＣ用外部関数(「Ｃユーザーズマニュアル」を参照)、ライブラリ(「Ｃライブラリマニュアル１」を参照)をご利用になる場合は、ＯＰＭＤＲＶ２．Ｘの環境をインストールしてお使いください。

なお、本書に記載のＯＰＭＤＲＶ３．Ｘ用のものとＯＰＭＤＲＶ２．Ｘ用のものを同時に利用することはできません。

CONTENTS

第１章　ミュージック …… １

１．１　準備 …… ２

１．２　機能説明 …… ３

１．３　オプションスイッチ …… ４

１．４　拡張ＭＭＬ書式 …… ６

１．５　サンプルプログラム …… ７

１．６　Ｘ－ＢＡＳＩＣ一覧 …… ８

１．７　Ｘ－ＢＡＳＩＣ関数リファレンス …… １０

１．８　ライブラリ一覧 …… ４５

１．９　ライブラリリファレンス …… ４７

１．１０　ＩＯＣＳ一覧 …… ９０

１．１１　ＩＯＣＳコールリファレンス …… ９２

１．１２　ＯＰＭコマンド …… １３４

１．１３　エラーコード一覧 …… １３９

第２章　ＳＣＳＩ …… １４１

２．１　サンプルプログラム …… １４２

２．２　ライブラリ一覧 …… １４４

２．３　ライブラリリファレンス …… １４５

２．４　ＩＯＣＳ一覧 …… １５８

２．５　ＩＯＣＳコールリファレンス …… １５９

２．６　エラーコード …… １７２

第１章　　ミュージック

「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ　ｖｅｒ２．１」では新たにＦＭ音源／ＭＩＤＩ／ＡＤＰＣＭを同時にコントロールできる機能が追加されています。

これにより、以下のファイルが追加／変更されています。

ファイル名	内　　　容	ディレクトリ
MUSIC.DEF	FM音源（MIDI/ADPCM対応）用ＤＥＦファイル	ｼｽﾃﾑ2の¥BC
MUSIC3.DEF	FM音源/MIDI/ADPCM用拡張ＤＥＦファイル	ｼｽﾃﾑ2の¥BC
AUDIO.FNC	ＡＤＰＣＭ外部関数	ｼｽﾃﾑ1の¥BASIC2
MUSIC.FNC	FM音源（MIDI/ADPCM対応）用外部関数	ｼｽﾃﾑ1の¥BASIC2
MUSIC3.FNC	FM音源/MIDI/ADPCM用拡張外部関数	ｼｽﾃﾑ1の¥BASIC2
MUSIC.H	FM音源（MIDI/ADPCM対応）用関数の定義	ｼｽﾃﾑ2の¥INCLUDE
MUSIC3.H	FM音源/MIDI/ADPCM用拡張関数の定義	ｼｽﾃﾑ2の¥INCLUDE
BASLIB.L	ＢＡＳＴＯＣライブラリ	ｼｽﾃﾑ2の¥LIB
IOCSCALL.MAC	ＩＯＣＳコールのヘッダファイル	ｼｽﾃﾑ2の¥INCLUDE

これらのファイルを利用して作成したプログラムを動作させるためには、ＯＰＭＤＲＶ３．Ｘが必要です。
なお、従来のＯＰＭＤＲＶ２．Ｘ用のファイルは、システムディスク２のＭＵＳＩＣディレクトリに入っています。

1.1 準備

新しいミュージックの機能を使用するためには、次のようにCONFIG.SYS、BASIC.CNFを設定してください。
なお、以下の設定をフッロピーディスクでは、オート・インストール（INSTALL.BAT）で自動的に行なえます。また、ハードディスクでは、オート・インストールでCONFIG.SYSの登録以外は自動的に行なえます。（Cユーザーズマニュアル参照）

● CONFIG.SYS

1）SYSディレクトリにOPMDRV3.X をコピーしてください。
2）CONFIG.SYS に 次のようにOPMDRV3.X を登録してください。

CONFIG.SYSの例

```
FILES     = 15
BUFFERS   = 20 1024
LASTDRIVE = Z:
KEY       = ¥KEY.SYS
USKCG     = ¥USKCG.SYS
BELL      = ¥BEEP.SYS
DEVICE    = ¥SYS¥PRNDRV.SYS
DEVICE    = ¥SYS¥ASK68K.SYS B:¥X68K_M.DIC B:¥X68K_S.DIC ¥ASK¥ENV1.ASK
DEVICE    = ¥SYS¥FLOAT2.X
DEVICE    = ¥SYS¥OPMDRV3.X #180 /P64 /OPM
DEVICE    = ¥SYS¥SCSIDRV.SYS /ID0
DEVICE    = ¥SYS¥RAMDISK.SYS #G
DEVICE    = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
DEVICE    = ¥SYS¥IOCS.X
ENVSET    = 512 ¥STARTUP.ENV
PROCESS   = 5 2 20
```

注）PCMDRV.SYSの機能は、OPMDRV3.Xに含まれていますので登録する必要はありません。

● BASIC.CNF

1）BASIC2ディレクトリに新しく追加／変更された
AUDIO.FNC、MUSIC.FNC、MUSIC3.FNC
をコピーしてください。
2）BASIC.CNF にAUDIO、MUSIC、MUSIC3を追加定義してください。

BASIC.CNFの例

```
FREE  = 128
WIDTH = 96
BEEP  = ON
CAPS  = OFF
FUNC  = GRAPH
FUNC  = MOUSE
FUNC  = SPRITE
FUNC  = STICK
FUNC  = AUDIO
FUNC  = MUSIC
FUNC  = MUSIC3
```

1.2 機能説明

ＯＰＭＤＲＶ３．Ｘでは、ＦＭ音源／ＭＩＤＩ／ＡＤＰＣＭを次のように同時にコントロールできます。
なお、ＯＰＭＤＲＶ３．ＸはＯＰＭＤＲＶ．Ｘの上位互換になっていますが、ＯＰＭＤＲＶ２．Ｘとは互換性がありませんのでご注意ください。

- 以下のデバイス名をサポートしています。

PCM	：ＰＣＭの録音／再生
MIDI、OPM	：MMLデータの再生、及び各コマンドの実行
MIDIA	：アスキーファイルをMIDIコードに変換してMIDI OUTへ出力
MIDIB	：バイナリーファイルをそのままMIDIデータとして出力
MIDIAE	：アスキー エクスクルーシブデータ ファイルをMIDIコードに変換して出力（Musicstudio PRO-68Kで作成されるＥＸＣファイルなど）
MIDIBE	：バイナリー エクスクルーシブデータ ファイルをそのままMIDIデータとして出力（ミュ-1で作成されるＥＸＢファイルなど）

注）エクスクルーシブデータを出力する場合のデバイス名は、'MIDIAE'または'MIDIBE'で行ってください。
（'F0''F7'のチェックを行い、受信側がオーバーフローをおこさないようにウェイトを挿入します。）

- ＦＭ音源／ＭＩＤＩ／ＡＤＰＣＭを任意のチャンネル(1～25)に割り当てて鳴らす事ができます。
- ＡＤＰＣＭ音を１２８種類まで登録することができます。
（ＭＭＬ中では任意のノートに割り当てて記述できます。）
- トラック数は８０で、２５トラックまで同時演奏することができます。
- １トラックで和音を演奏することができます。（ＭＩＤＩのみ）
- モジュレーション／ピッチベンドのコマンドは、そのままＦＭ音源に対しても機能します。
- 常にＭＩＤＩクロック(F8)を出力しているので、外部シーケンサーを同期演奏させることができます。
外部同期に設定すれば、外部シーケンサーでコントロールすることもできます。
任意の小節でのスタート／ストップが可能です。
- SOUND PRO-68KのＳＮＤファイルで音色設定ができます。
（音色名もそのまま登録されます。）
- ＭＭＬデータの中にチャンネル変更コマンドを自由に設定できます。
- MUSIC PRO-68K[MIDI]のＭＵＳファイルをそのまま演奏できます。
（オプションスイッチ／Ｙを設定してください。）
- ＯＰＭＤＲＶ３．Ｘ上でMUSIC PRO-68K、MUSIC PRO-68K[MIDI]がそのまま利用できます。
（オプションスイッチ／Ａを設定してください。）

1．3 オプションスイッチ

ＯＰＭドライバ３（ＯＰＭＤＲＶ３．Ｘ）では登録時に以下のオプションスイッチを設定できます。

スイッチ	内　　　容
#n	ｎ にトラックバッファのサイズを１Ｋバイト単位で指定します。 指定範囲は24kﾊﾞｲﾄ～4096kﾊﾞｲﾄで、省略した場合は６４Ｋバイトを確保します。 （例: OPMDRV3 #120 ）
/Pn	ｎ にＰＣＭデータのバッファサイズをＫバイト単位で指定します。 指定範囲は0kﾊﾞｲﾄ～4096kﾊﾞｲﾄで省略した場合は２４Ｋバイトを確保します。 ＡＤＰＣＭを使用しない場合は ｎ に 0 を指定してください。 （例: OPMDRV3 /P200 ）
/Yn	MMLデータの Yコマンドのモードを指定します。デフォルトは/Y1に設定されます。 （例: OPMDRV3 /Y0 /MOP) MUSIC PRO-68K[MIDI]のMUSﾌｧｲﾙを演奏時の設定
/OMP	各音源に対するチャンネル番号を指定します。 省略した場合は、/OPMの設定になります。 （例: OPMDRV3 /OMP ） O=OPM　（使用チャンネル数８　：ch=１～８） M=MIDI　（使用チャンネル数１６：ch=９～２４） P=PCM　（使用チャンネル数１　：ch=２５） アルファベットで指定された順に、各音源で使用するチャンネル番号を割り当てます。 上記の場合は、OPM=ch1～8 MIDI=ch9～24 PCM=ch25 がそれぞれの音源に対するチャンネル番号となります。 MIDIを使用しない場合は /OPM と指定しておけば、従来のＭＭＬデータをそのまま演奏できます。 (OPM=ch1～8　PCM=ch9　MIDI=ch10～25) MIDIに指定されたchは、その範囲を MIDIch1～16 にシフトしてMIDIOUTに出力されます。'/OPM'でch10～25 が指定されている場合は、ch10 がMIDIch1 に、ch25 が MIDIch16になります。 これらの設定は、起動後にも変更することができます。 (BASIC上では m_sysch() コマンドで、ＯＳ上では(/OMP)などで変更)

スイッチ	内　　　　　　容
/Fn	ＦＭ音源全体の音程を微調整します。(n=0～63) ﾃﾞﾌｫﾙﾄは0
/R	OPMDRV3.Xを解除します。
/A	MUSIC PRO-68K[MIDI]を使用する時に設定します。

● 各スイッチを省略した場合のデフォルトは #64 /OPM /Y1 /P24 となります。

● ＭＵＳファイルを演奏する場合は次の様に指定してください。

```
                                    ┌─ＭＩＤＩコントロールデータ出力モード
DEVICE = ¥SYS¥OPMDRV3.X #120 /MOP /Y0
```

尚、それぞれの曲データの先頭に以下のＯＰＭコマンドを追加記入しておけば、
上記の選択は不要になります。（ＯＰＭコマンドを参照）

OPMファイルを演奏する場合 (/OPM)(/Y1)

MUSファイルを演奏する場合 (/MOP)(/Y0)

1.4 拡張MML書式

ＯＰＭＤＥＶ３．Ｘでは次のＭＭＬが拡張されています。その他のＭＭＬにつきましては、本体付属の「Ｘ－ＢＡＳＩＣ　ユーザーズリファレンス」をご覧ください。
（ＯＰＭＤＲＶ２．Ｘで拡張されたＭＭＬは使用できません。）

MMLデータ	意　　　味	パラメータ範囲（初期値）
@Nn	チャンネル変更	:n=1～25
@Pn	パンポットの設定	:n=0(右)..127(左)
@Vn	ボリュームの設定	:n=0～127
@Un	ベロシティの設定	:n=0～127
@Dn	ダンパーオンオフの設定	:n=0(off)　1(on)
@Bn	ピッチベンドの設定 type 1	:n=0～8192～16383（半音≒683)
@Bn1,n2	ピッチベンドの設定 type 2	:n1=0～127(low byte) :n2=0～127(hi byte)
@Mn1,n2	モジュレーションの設定 n2を省略した場合は198 (speedはＦＭ音源のみに有効)	:n1=0～127(depth) :n2=0～255(speed)
' '	和音の指定:最初の音程のみ長さを指定可:異なる長さの音は混在不可。 最大８音（ > < # - + は設定可) （ドラムのパートなどを１トラックですますことができます。） ＭＩＤＩ以外のチャンネルでは和音演奏はできません。 例 'C8EG<C>'	

注）

MUSIC PRO-68K [MIDI] のＭＵＳファイルを演奏する場合は必ず、(/MOP)(/YO)のモードで行ってください。

m_alloc関数または(M)コマンドで確保するメモリサイズは従来の１．２倍位に設定してください。

もし演奏がおかしくなる場合は、m_trk関数実行後にm_use/m_free 関数でメモリ 使用状況を確認し、新たに設定しなおしてください。

1.5 サンプルプログラム

```
100 /*
110 /* music sample
120 /* "おお、スザンナ！！"     S.C.Foster
130 /*
140 str tr,as,bs,cs,ds,part[255]
150 /*
160 m_init()
170 m_sysch("OPM")
180 m_alloc(10,2000)
190 m_assign(10,10)
200 /*
210 linput "演奏チャンネルを指定してください           －－－－－－－＞？";tr
220 linput "音色番号を入力してください（１～２００）－－－＞？";as
230 linput "テンポを入力してください（２０～３００）   －－－－－－－＞？";bs
240 print
250 /*
260 m_trk(10,"@N"+tr+"@"+as+" o4 v9 q7 l16 t"+bs)
270 m_trk(10,"g&a b8<d8d8e8> <d8>b8g8.a b8b8a8g8 a4.ga&")
280 m_trk(10,"b8<d8d8.e> <d8>b8g8.a& b8b8a8a8 g4.ga&")
290 m_trk(10,"b8<d8d8.e> <d8>b8g8.a b8b8a8g8 a4r8ga&")
300 m_trk(10,"b8<d8d8e8> <d8.>bg8.a bb8.a8.a g4r4")
310 m_trk(10,"<c4c4> <e8e4e8> <d8.d>b8g8 a4r8ga&")
320 m_trk(10,"b8<d8d8.e> <d8>b8g8a8 b8b8a8.a g4r8")
330 /*
340 m_play()
350 while (m_stat() <> 0)
360     input "演奏を終了しますか (Y/N) ?",cs
370         if ((cs = "Y") or (cs = "y")) then {
380             m_stop() : end
390         } else {
400             m_cont()
410         }
420 endwhile
430 end
```

1.6 X-BASIC一覧表

番号	名　　前	機　　能	頁
$00	M_INIT	トラックバッファの初期化	17
$01	M_ALLOC	指定のトラックのトラックバッファの確保	11
$02	M_ASSIGN	指定チャンネルのトラック設定	12
$03	M_VGET	ＦＭ音源の音色データ取得	41
$04	M_VSET	ＦＭ音源の音色データを登録	43
$05	M_TEMPO	テンポ指定	37
$06	M_TRK	指定トラックにＭＭＬデータをセット	39
$07	M_FREE	指定トラックバッファの残りバイト数を返す	16
$08	M_PLAY	指定チャンネルの演奏開始	31
$09	M_STAT	指定チャンネルの演奏状態を返す	35
$0A	M_STOP	指定チャンネルの演奏を一時停止	35
$0B	M_CONT	指定チャンネルの演奏再開	13
$0F	M_USE	指定トラックバッファの使用容量を返す	40
$10	M_CHAN	データ出力チャンネルを一時変更	13
$11	M_PROG	指定チャンネルの音色番号設定	33
$12	M_VOL	指定チャンネルにボリューム設定	41
$13	M_PAN	指定チャンネルのパンポット指定	24
$14	M_VEL	指定チャンネルのベロシティ設定	40
$15	M_TRNS	指定チャンネルの音程シフト	39
$18	M_ANTOFF	指定チャンネルへのオールノートオフ出力	11
$19	M_DIROUT	ＭＩＤＩ出力	14
$1B	M_NTON	指定チャンネルへノートオン出力	22
$1C	M_NTOFF	指定チャンネルへノートオフ出力	21
$20	M_SYNC	ＭＩＤＩの同期モードを設定	36
$21	M_METER	曲の拍子設定	19
$29	M_SNDSET	SNDファイルのデータをＦＭ音源用に登録	33
$2A	M_PCMSET	指定の音程にＰＣＭ音を登録	30
$2B	M_SYSCH	チャンネル番号の割り当て変更	37
$2C	M_IFCHK	ＭＩＤＩインターフェイスの有無チェック	17
$2F	M_FREEA	ﾄﾞﾗｲﾊﾞ起動時のﾄﾗｯｸﾊﾞｯﾌｧの全ﾊﾞｲﾄ数を返す	16
$30	M_OUT	1バイトデータをＭＩＤＩへ出力	24
$32	M_MEAS	演奏中の小節番号を返す	18
$33	M_YCOM	Ｙコマンドの解釈	44
$34	M_START	演奏開始小節の設定	34
$35	M_END	演奏停止小節の設定	15
$36	M_MODSNS	ＦＭ音源のモジュレーションの深さ設定	20
$38	M_PCMREC	外部入力音のＰＣＭ録音	29
$39	M_PCMGET	登録されているＰＣＭデータの取得	26
$3A	M_PCMLEN	指定されたＰＣＭデータのサイズを返す	27
$3B	M_PCMBSY	ＰＣＭの実行状態を返す	25
$3C	M_PCMON	指定されたＰＣＭデータの再生	28
$3D	M_CTRLRES	指定チャンネルへの各種のオフ出力	14
$3E	M_OPMREG	指定のＯＰＭレジスタにデータをセット	23
$3F	M_MDREG	ＭＩＤＩ制御レジスタにデータをセット	18
$40	M_TNMSET	ＦＭ音源の音色名の登録	38
$41	M_TNMGET	ＦＭ音源の音色名の取得	38
$42	M_PNMSET	ＰＣＭ音色名の登録	32
$43	M_PNMGET	ＰＣＭ音色名の取得	32
$44	M_MSTVOL	マスターボリュームの設定	20

番号	名　　前	機　　能	頁
$45	M_BEND	指定チャンネルのピッチベンド設定	12
$46	M_MOD	指定チャンネルのモジュレーション設定	19
$47	M_OPMEXC	ＦＭ音源のパラメータセットの選択	22
$48	M_OPMLFQ	ＦＭ音源のＬＦＯの発振周波数の設定	23
$49	M_PGMFLT	ＭＭＬ音色切り替えコマンドの出力選択	31
$4A	M_VOLFLT	ＭＭＬボリューム設定コマンドの出力選択	42
$4B	M_PANFLT	ＭＭＬパンポット設定コマンドの出力選択	25
	M_ERRGET	直前のMUSIC関数のエラーコードを返す	15
	M_MUTE	指定チャンネルの音を消音	21
	M_PCMCLR	ＰＣＭバッファのクリア	26
	M_SOLO	指定チャンネル以外の音を消音	34

1.7 X-BASIC関数リファレンス

「C compiler PRO−68K ver2.1」では、OPMDRV.X用に用意されたX−BASIC外部関数の上位互換な（FM音源／MIDI／ADPCMを同時に制御できる）OPMDRV3.X用のX−BASIC外部関数がサポートされています。
この拡張されたX−BASIC外部関数はXBASTOCなどにより実行ファイルにコンパイルすることも可能です。このため、新しく次のファイルがver2.1より追加／変更されています。

ファイル名	内　容
MUSIC.DEF	FM音源（MIDI/ADPCM対応）用ＤＥＦファイル
MUSIC3.DEF	FM音源/MIDI/ADPCM用拡張ＤＥＦファイル
AUDIO.FNC	ＡＤＰＣＭ外部関数
MUSIC.FNC	FM音源（MIDI/ADPCM対応）用外部関数
MUSIC3.FNC	FM音源/MIDI/ADPCM用拡張外部関数
MUSIC.H	FM音源（MIDI/ADPCM対応）用関数の定義
MUSIC3.H	FM音源/MIDI/ADPCM用拡張関数の定義
BASLIB.L	ＢＡＳＴＯＣライブラリ

なお、「C compiler PRO−68K ver2.0」でサポートされていたOPMDRV2.X用のX−BASIC外部関数とは互換性がありませんのでご注意ください。

従来のOPMDRV2用のファイルは、システムディスク2のMUSICディレクトリに入っています。「Cユーザーズマニュアル」に記載のX−BASIC関数をご利用になる場合は、そのファイルをご使用ください。
（OPMDRV3.X用の関数と同時に利用することはできません）

従来PCMDRV.SYSでサポートされていた機能もOPMDRV3.Xで包含していますので、このX−BASIC関数リファレンスに記載されていないADPCM用のX−BASIC外部関数（A_PLAY、A_REC）も従来通りお使いいただけます。
詳細は、本体付属の「X−BASICユーザーズリファレンス」を参照してください。

M_ALLOC

MUSIC

書　式	m_alloc([t] ,s)
引　数	char(t),int(s)
戻り値	int
機　能	指定されたトラックtのトラックバッファを確保します。

t・・・・・・・・トラック番号(1～80)
s・・・・・・・・バッファサイズ(1～65536)

全トラックバッファ内のデータはクリアされます。
また、バッファサイズに1～3を指定しても、4バイト確保します。
tを省略すると全てのトラックをバッファサイズSで確保します。
なお、総トラックのバッファサイズ（それぞれのトラックに割り当てられているトラックバッファの総合計）は、ＣＯＮＦＩＧ．ＳＹＳファイル内で指定してください。

DEVICE = ¥SYS¥OPMDRV3.X #nnn
(nnn：総トラックバッファサイズ［単位：ＫＢ］）

戻り値として、正常のときは0、エラーのときは-1を返します。

M_ANTOFF

MUSIC3

書　式	m_antoff([c])
引　数	char
戻り値	int
機　能	指定のチャンネルcへオールノートオフを出力します。

c・・・・・・・・チャンネル番号 (1～25)

cが省略された場合、全ての チャンネルへオールノートオフを出力します。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_ASSIGN

MUSIC

書式	m_assign(c [,t])
引数	char
戻り値	int
機能	指定のチャンネルcで使用するトラックtを設定します。

c・・・・・・・・チャンネル番号(1～25)
t・・・・・・・・トラック番号(1～80)

また、tが省略された場合は、チャンネルcに設定されているトラック番号を返します。
複数のチャンネルに同一のトラックを割りあてると、それぞれ同じ楽譜データを演奏します。また、複数のトラックを1つのチャンネルに割りあてると、最も新しく割りふられたトラックの楽譜データを演奏します。
デフォルト値は、チャンネル1～8が、それぞれトラック1～8に対応します。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_BEND

MUSIC3

書式	m_bend([c] ,b)
引数	char
戻り値	int
機能	指定のチャンネルのピッチベンドを設定します。

c・・・・・・・・・・・チャンネル番号(1～25)
b・・・・・・・・・・・ピッチベンドデータ(0～127)

cが省略された場合は、全てのチャンネルにピッチベンド b を設定します。bは省略できません。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_CHAN

MUSIC3

書　式	m_chan(c1 [,c2])
引　数	char
戻り値	int
機　能	指定のチャンネルのデータ出力チャンネルを一時的に変更します。

c1・・・・・・・チャンネル番号(1～25)
c2・・・・・・・出力チャンネル番号(1～25)

c2 が省略された場合は、現在の設定値を返します。
m_assign関数やMMLデータ @N とは異なり、チャンネル変更前のパラメータ(トラック番号、音色番号、ボリュームなど)を引き継ぎます。
m_init関数実行まで有効です。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_init

M_CONT

MUSIC

書　式	m_cont([c1] [,c2] [,c3] [,c4] [,c5] [,c6] [,c7] [,c8])
引　数	char
戻り値	int
機　能	c1からc8で指定されたチャンネルにおいて、一時停止した演奏を再開します。

c1～c8・・・・・・・・チャンネル番号(1～25)

c1からc8をすべて省略すると、現在一時停止しているすべてのチャンネルの演奏を再開します。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_stop

M_CTRLRES

書　式	m_ctrlres([c])
引　数	char
戻り値	int
機　能	指定のチャンネル c ヘピッチベンドオフ／モジュレーションオフ／ダンパーオフ／オールノートオフを出力します。 c・・・・・・・・チャンネル番号(1～25) cが省略された場合、全てのチャンネルへピッチベンドオフ／モジュレーションオフ／ダンパーオフ／オールノートオフを出力します。 戻り値として、正常のときは0、エラーのときは-1を返します。

M_DIROUT

書　式	m_dirout(lng,ca)
引　数	int(lng),char型一次元配列名(ca)
戻り値	int
機　能	lngバイトのＭＩＤＩデータをＭＩＤＩへ出力します。 lng・・・・・・lngのとりうる値は1から配列名caで宣言された配列の添字+1までです。 ca・・・・・・・ＭＩＤＩデータを格納しているchar型一次元配列名

➡ m_out

M_END

MUSIC3

書　式	m_end(m)
引　数	int
戻り値	int
機　能	演奏を停止する小節を設定します。 　　　m・・・・・・・・・演奏を中止する小節番号 全てのチャンネルの"|:","D.S","D.C",":|"などの位置が同じでないと正常な演奏ができません。 戻り値として、正常のときは0、エラーのときは-1を返します。

➡　m_start

M_ERRGET

MUSIC3

書　式	m_errget()
戻り値	int
機　能	直前のＭＵＳＩＣ関数のエラーコードを返します。 正常終了している場合は０を返し、エラーの場合は１～６５の エラーコードを返します。エラーコードについては、「１．１３エラーコード一覧」を参照してください。

M_FREE

MUSIC

書　式	m_free(t)
引　数	char
戻り値	int
機　能	指定のトラックtのトラックバッファの残りバイト数を返します。 t·········トラック番号(1～80)

➡ m_freea,m_use

M_FREEA

MUSIC3

書　式	m_freea()
戻り値	int
機　能	ドライバ起動時に確保したトラックバッファの全バイト数を返します。

➡ m_free,m_use

M_IFCHK

MUSIC3

書式	m_ifchk()
戻り値	int
機能	MIDIインターフェイスの有無をチェックします。 インターフェイスがつながっている場合は 0、つながっていない場合は -1を返します。

M_INIT

MUSIC

書式	m_init(md)
引数	char(md)
戻り値	int
機能	トラックバッファを初期化します。各トラックバッファは4バイトに設定されます。 また、各チャンネルのパラメータ（トラック番号、音色番号、ボリュームなど）も初期化されます。 mdに1を指定した場合、FM音源の音色データも初期化します。 mdに0設定または省略した場合は音色データは初期化しません。 md・・・・・・・初期化モード 0 = トラックバッファの初期化 1 = トラックバッファ、音色データの初期化

M_MDREG

MUSIC3

書　式	m_dreg(gr,adr,dt)
引　数	char
戻り値	int
機　能	ＭＩＤＩ制御ＩＣ（ＹM3802）のレジスタに値を書き込みます。grにはグループナンバー、adrにはアドレスナンバーを指定します。dtにはレジスタに書き込む値を0～255の範囲で指定します。 戻り値として、正常のときは0、エラーのときは1を返します。 注：間違ったパラメータを指定すると、システムが正常に動作しなくなる場合があります。 ＹM3802の機能を十分に理解の上でご使用ください。

M_MEAS

MUSIC3

書　式	m_meas()
戻り値	int
機　能	現在演奏中の小節番号を返します。 ＭＭＬデータ"|:"," D.S." などのくり返し記号の場合はくり返し回数がそのまま加算されますので、実際の小節番号とは異なる場合があります。

➡ m_end,m_start

M_METER

MUSIC3

書　式	m_meter(n,d)
引　数	char
戻り値	int
機　能	曲の拍子を設定します。 n・・・・・・・・・分子(2～16) d・・・・・・・・・分母(2,4,8,16) デフォルト値は4/4拍子に設定されています。 戻り値として、正常のときは0、エラーのときは-1を返します。

M_MOD

MUSIC3

書　式	m_mod(〔c〕,m)
引　数	char
戻り値	int
機　能	指定のチャンネル c のモジュレーションを設定します。 c・・・・・・・・・チャンネル番号(1～25) m・・・・・・・・・モジュレーションデータ(0～127) cが省略された場合は、全てのチャンネルにモジュレーション m を設定します。 mは省略できません。 戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_modsns

M_MODSNS

MUSIC3

書　　式	m_modsns([m])
引　　数	char
戻 り 値	int
機　　能	FM音源全体のモジュレーションの深さを設定します。

m････････モジュレーションの深さ(0～127)

mが省略された場合は、現在の設定値を返します。

戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_mod

M_MSTVOL

MUSIC3

書　　式	m_mstvol([vol])
引　　数	char
戻 り 値	int
機　　能	マスターボリュームを設定します。

vol･･････ボリュームデータ(0～127)

デフォルト値は127

volが省略された場合、現在の設定値を返します。

m_init関数を実行するまで有効です。

ＰＣＭのチャンネルに対しては無効です。

戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_init,m_vol

M_MUTE

MUSIC3

書式	m_mute([c1] [,c2] [,c3] [,c4] [,c5] [,c6] [,c7] [,c8])
引数	char
戻り値	int
機能	c1からc8で指定したチャンネルの音をミュート(消音)します。 ノート以外のデータは常に出力されます。 c1～c8・・・・・・・・チャンネル番号(1～25) c1からc8を省略するとミュートが解除されます。 戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_solo

M_NTOFF

MUSIC3

書式	m_ntoff(c,nt [,vel])
引数	char
戻り値	int
機能	指定のチャンネルへノートオフを出力します。 c・・・・・・・・・チャンネル番号(1～25) nt・・・・・・・・ノート番号(0～127) vel・・・・・・・ベロシティデータ(0～127) velが省略された場合、ベロシティ値を６４で出力します。 戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_nton

M_NTON

MUSIC3

書　式	m_nton(c,nt, 〔vel〕)
引　数	char
戻り値	int
機　能	指定のチャンネルへノートオンを出力します。

c・・・・・・・・・チャンネル番号(1～25)
nt・・・・・・・・ノート番号(0～127)
vel・・・・・・・ベロシティデータ(0～127)

velが省略された場合、ベロシティ値を６４で出力します。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_ntoff

M_OPMEXC

MUSIC3

書　式	m_opmexc(〔md〕)
引　数	char
戻り値	int
機　能	ＦＭ音源にセットする音色パラメータの内、ＬＦＱ／ＰＭＤ／ＡＭＤなどの全体に影響するパラメータを音色切り替え時にセットするかどうかを選択します。

md・・・・・・・・・パラメータの設定スイッチ
0=セットしない
1=セットする　　（デフォルト値は1）

mdが省略された場合、現在の状態を返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_vset

M_OPMLFQ

MUSIC3

書　式	m_opmlfq([spd])
引　数	char
戻り値	int
機　能	ＦＭ音源のＬＦＯの発振周波数（スピード）を設定します。

spd・・・・・・・スピード(0～255、デフォルト値は198)

spdが省略された場合、現在のデータを返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_OPMREG

MUSIC3

書　式	m_opmreg(reg [,data])
引　数	char
戻り値	int
機　能	指定のＯＰＭレジスタにデータをセットします。

reg・・・・・・・ＯＰＭレジスタ番号(0～255)
data・・・・・・ＯＰＭへ書き込むデータ

dataが省略された場合、現在のregで示すＯＰＭレジスタの内容を返します。
一度も指定のレジスタが書き換えられていない場合は、－１を返します。
戻り値として、正常なときは0、エラーのときは-1を返します。

M_OUT

MUSIC3

書　式	m_out(data)
引　数	char
戻り値	int
機　能	1バイトデータをＭＩＤＩへ出力します。 data……ＭＩＤＩデータ(0～255) 戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_dirout

M_PAN

MUSIC3

書　式	m_pan(〔c〕〔,pan〕)
引　数	char
戻り値	int
機　能	指定のチャンネルcにのパンポットpanを設定します。 c………チャンネル番号(1～25) pan……パンポットデータ(0～127) デフォルト値は64 0 =ライト（右端） ⋮ 64 =センター（中央） ⋮ 127 =レフト（左端） cが省略された場合は、全てのチャンネルにpanを設定します。 panが省略された場合は、現在の設定値を返します。 戻り値として、正常なときは0、エラーのときは-1を返します。

M_PANFLT

書　式	m_panflt(〔c〕 〔,md〕)
引　数	char
戻り値	int
機　能	ＭＭＬデータのＰＡＮの切り替えコマンドPnを、演奏時に出力するかどうかを選択します。

c・・・・・・・・・チャンネル番号(1～25)
md・・・・・・・・0=Pnを出力する
　　　　　　　　1=Pnを出力しない

cが省略された場合は、全てのチャンネルにmdを設定します。
mdが省略された場合は、現在の設定値を返します。
戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_pgmflt、m_volflt

M_PCMBSY

書　式	m_pcmbsy()
戻り値	int
機　能	ＰＣＭの実行状態を調べます。

0x00 ： 何も実行していない
0x02 ： 出力中 (ADPCMOUT,m_pcmon,a_playを実行中)
0x04 ： 入力中 (ADPCMINP,m_pcmrec,a_recを実行中)
0x12 ： 出力中 (ADPCMAOTを実行中)
0x14 ： 入力中 (ADPCMAINを実行中)
0x22 ： 出力中 (ADPCMLOTを実行中)
0x24 ： 入力中 (ADPCMLINを実行中)

M_PCMCLR

MUSIC3

書　式	m_pcmclr()
戻り値	int
機　能	ＰＣＭバッファをクリアします。 m_pcmset関数などで登録したＰＣＭデータが失われます。 戻り値として、正常なときは0、エラーのときは-1を返します。

M_PCMGET

MUSIC3

書　式	m_pcmget(nt,ca)
引　数	char(nt),char型一次元配列名(ca)
戻り値	int
機　能	登録されているＰＣＭデータをcaに読み込みます。 nt・・・・・・・・ノート番号 ca・・・・・・・・ＰＣＭデータバッファを格納するためのchar型一次元配列名 戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_pcmlen,m_pcmset

M_PCMLEN

書　式	m_pcmlen([nt])
引　数	char
戻り値	int
機　能	ノート番号ntに登録されているＰＣＭデータのサイズを返します。 　　nt・・・・・・・ノート番号(0～127) ntを省略すると、ＰＣＭバッファの残り容量を返します。 戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_pcmset,m_pcmget

M_PCMON

書　式	m_pcmon(nt〔,sf〕〔,md〕)
引　数	char
戻り値	int
機　能	指定されたＰＣＭデータを再生します。

nt・・・・・・・・ノート番号(0～127)

sf・・・・・・・・サンプリング周波数

0・・・・・・3.9KHz
1・・・・・・5.2KHz
2・・・・・・7.8KHz
3・・・・・・10.4KHz
4・・・・・・15.6KHz

音声を再生する場合、原則としてm_pcmrec関数で録音したときのサンプリング周波数sfを使用してください。それ以外を使用すると、録音した音声を正常に再生できません。

md・・・・・・・・音声出力モード（オーディオ出力端子）

0・・・・・・カット
1・・・・・・左
2・・・・・・右
3・・・・・・ステレオ（左右両方）

sf,mdが省略された場合は sf=4 md=3で再生します。
戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_pcmrec,m_pcmset

M_PCMREC

書　式	m_pcmrec(sf,lng,ca)
引　数	char(sf),int(lng),char型一次元配列名(ca)
戻り値	int
機　能	外部入力音（オーディオ入力端子）をＰＣＭ録音します。 実行直後に入力待ちになり、約４秒以内に入力がなければ 実行を中止します。 sf・・・・・・・・サンプリング周波数 0・・・・・・3.9KHz(１秒間に消費するメモリサイズ1950バイト) 1・・・・・・5.2KHz(１秒間に消費するメモリサイズ2600バイト) 2・・・・・・7.8KHz(１秒間に消費するメモリサイズ3900バイト) 3・・・・・・10.4KHz(１秒間に消費するメモリサイズ5200バイト) 4・・・・・・15.6KHz(１秒間に消費するメモリサイズ7800バイト) lng・・・・・・・録音する配列caの添字0からの長さ(最大=32767) lngのとりうる値は、１から、配列名caで宣言された配列の添字＋１までです。 ca・・・・・・・・ＰＣＭを格納するchar型一次元配列名 戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_pcmon,m_pcmset

M_PCMSET

書　式	m_pcmset(nt,sf,lng,ca)
引　数	char(nt,sf),int(lng),char型一次元配列名(ca)
戻り値	int
機　能	指定の音程にＰＣＭ音を登録します。

指定できる音程は'C-2'(0)から'G#8'(127)までです。

nt・・・・・・・・ノート番号(0～127)

sf・・・・・・・・サンプリング周波数

0・・・・・・3.9KHz

1・・・・・・5.2KHz

2・・・・・・7.8KHz

3・・・・・・10.4KHz

4・・・・・・15.6KHz

lng・・・・・・・登録する配列caの添字0からの長さ(最大=32767)

lngのとりうる値は、１から、配列名caで宣言された配列の添字＋１までです。

ca・・・・・・・・ＰＣＭデータが格納されている一次元配列名

戻り値として、正常なときは0、エラーのときは-1を返します。

➡　m_pcmget,m_pcmlen,m_pcmon,m_pcmrec

M_PGMFLT

MUSIC3

書式	m_pgmflt([c] [,md])
引数	char
戻り値	int
機能	MMLデータの音色の切り替えコマンド@nを、指定チャンネルの演奏時に出力するかどうかを選択します。

c・・・・・・・・チャンネル番号(1～25)
md・・・・・・・0=@nを出力する。
　　　　　　　1=@nを出力しない。

cが省略された場合は、全てのチャンネルに md を設定します。
mdが省略された場合は、現在の設定値を返します。
戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_panflt,m_volflt

M_PLAY

MUSIC

書式	m_play([c1] [,c2] [,c3] [,c4] [,c5] [,c6] [,c7] [,c8])
引数	char
戻り値	int
機能	c1～c8で指定されたチャンネルに割り当てられたトラックの楽譜データを演奏します。

c1～c8・・・・チャンネル番号(1～25)

c1～c8をすべて省略すると、全チャンネルを演奏します。
戻り値として、正常なときは0、エラーのときは-1を返します。

M_PNMGET

書　式	m_pnmget(nt)
引　数	char
戻り値	str
機　能	ノート番号ntに登録されているＰＣＭの音色名を取得します。 nt・・・・・・・ノート番号(0～127)

➡ m_pnmset

M_PNMSET

書　式	m_pnmset(nt,pn)
引　数	char(nt),str(Pn)
戻り値	int
機　能	ノート番号ntにＰＣＭの音色名を登録します。(半角10文字以内) nt・・・・・・・ノート番号(0～127) pn・・・・・・・ＰＣＭの音色名 戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_pnmget

M_PROG

MUSIC3

書式	m_prog([c] [,p])
引数	char
戻り値	int
機能	指定のチャンネルcに音色番号pで示される音色を設定します。

c・・・・・・・・・チャンネル番号(1～25)
p・・・・・・・・・音色番号(FM音源は1～200、MIDIは1～128)

c が省略された場合は、全てのチャンネルに vo を設定します。
p が省略された場合は、現在の設定値を返します。
戻り値として、正常なときは0、エラーのときは-1を返します。

M_SNDSET

MUSIC3

書式	m_sndset(ca)
引数	char型一次元配列(ca)
戻り値	int
機能	SOUND PRO-68K のＳＮＤファイル形式のデータをＦＭ音源用のデータとして登録にします。同時に音色名も登録します。

ca ………… ＳＮＤファイル形式のデータ
１音色８０バイトで最大２００音色（=16000バイト）
まで指定できます。

戻り値として、正常のときは0、エラーのときは-1を返します。

M_SOLO

MUSIC3

書式	m_solo([c])
引数	char
戻り値	int
機能	指定されたチャンネルc以外の全ての音をミュートします。

c・・・・・・・・・チャンネル番号(1～25)

c を省略するか0を指定すると解除されます。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_mute

M_START

MUSIC3

書式	m_start(m)
引数	int
戻り値	int
機能	演奏を開始する小節を設定します。

m・・・・・・・・・演奏を開始する小節番号

m_play関数を実行することにより、小節番号mの小節より演奏を開始します。全てのチャンネルのMMLデータ“ |:”,“ D.S.”,“ D.C.”,“ :|”などの数や位置が同じでないと正常な演奏ができません。（一度設定された小節番号は演奏が終わるとクリアされます。）
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_end,m_meas,m_play

M_STAT

MUSIC

書　式	m_stat([c])
引　数	char
戻り値	int
機　能	指定のチャンネルが演奏中なら1を、停止中なら0を返します。

c･････････チャンネル番号(1～25)

cが省略された場合は、全てのチャンネルの状態をbit0～24に返します。bit0～24がチャンネル1～25に対応しています。それぞれのbitが0ならば停止中、1ならば演奏中です。

M_STOP

MUSIC

書　式	m_stop([c1] [,c2] [,c3] [,c4] [,c5] [,c6] [,c7] [,c8])
引　数	char
戻り値	int
機　能	c1～c8で指定されたチャンネルの演奏を一時停止します。

c1～c8････チャンネル番号(1～25)

チャンネルをすべて省略すると、全チャンネルの演奏を一時停止します。戻り値として、正常のときは0、エラーのときは-1を返します。

➡　m_cont

M_SYNC

書　式	m_sync([c])
引　数	char
戻り値	int
機　能	同期モードを設定します。

同期モード
c・・・・・・・・・0 内部のＭＩＤＩクロックに合わせて演奏
1 外部のＭＩＤＩクロックに合わせて演奏
2 外部のＦＳＫクロックに合わせて演奏

外部同期にする場合は、m_sync(1) 実行後 m_play関数で待機状態になり
ＭＩＤＩのスタートコマンド($FA)を待ちます。
cを省略すると、現在の状態を返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_SYSCH

MUSIC3

書　式	m_sysch([cm])
引　数	str
戻り値	int
機　能	各音源に対するチャンネル番号の割り当てを変更します。

(ドライバ組み込み時のオプションスイッチと同じ)

cm・・・・・・・各音源のチャンネルの割り当てを示す文字列

"OPM"・・・・・・・	OPM=ch1～8	PCM=ch9	MIDI=ch10～25
"OMP"・・・・・・・	OPM=ch1～8	MIDI=ch9～24	PCM=ch25
"MOP"・・・・・・・	MIDI=ch1～16	OPM=ch17～24	PCM=ch25
"MPO"・・・・・・・	MIDI=ch1～16	PCM=ch17	OPM=ch18～25
"POM"・・・・・・・	PCM=ch1	OPM=ch2～9	MIDI=ch10～25
"PMO"・・・・・・・	PCM=ch1	MIDI=ch2～17	OPM=ch18～25

cmを省略した場合は、現在のモードを数値で返します。

(0="OPM",1="OMP",2="MOP",3="MPO",4="POM",5="PMO")

戻り値として、正常のときは0、エラーのときは-1を返します。

M_TEMPO

MUSIC

書　式	m_tempo([te])
引　数	int
戻り値	int
機　能	テンポteを1分間に4分音符を何回打つかで指定します。

テンポは、20～300の範囲です。(デフォルト値は120)

teが省略された場合は、現在のテンポを返します。

戻り値として、正常なときは0、エラーのときは-1を返します。

M_TNMGET

MUSIC3

書　式	m_tnmget(vo)
引　数	char
戻り値	str
機　能	音色番号voで示される音色データに登録されている音色名を取得します。 vo・・・・・・・・音色番号(1～200) 戻り値として音色名を示す文字列を返します。（半角で10文字以内）

➡ m_tnmset

M_TNMSET

MUSIC3

書　式	m_tnmset(vo,tn)
引　数	char(vo),str(tn)
戻り値	int
機　能	音色番号voで示される音色データに音色名を登録します。 vo・・・・・・・・音色番号(1～200) tn・・・・・・・・音色名を示す文字列（半角10文字以内） 戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_tnmget

M_TRK

MUSIC

書　式	m_trk(t,st)
引　数	char(t),str(st)
戻り値	int
機　能	MMLによる楽譜データをトラックtに書き込みます。厳密にいうとトラックtに楽譜データを追加していきます。

t・・・・・・・・・トラック番号(1〜80)
st・・・・・・・・MML(ミュージックマクロランゲージ)データ

MMLデータの終端は、ヌル文字（0）です。
戻り値として、正常なときは0、エラーのときは-1を返します。

M_TRNS

MUSIC3

書　式	m_trns([c] [,stp])
引　数	char
戻り値	int
機　能	指定のチャンネルcの音程をシフトします。(半音ごとに±2オクターブの範囲)

cが省略された場合は、全てのチャンネルに stp を設定します。
stpが省略された場合は、現在の設定値を返します。

c・・・・・・・・・チャンネル番号
stp・・・・・・・音程を移調するデータ(0〜48)
　　　　　　デフォルト値は24

戻り値として、正常のときは0、エラーのときは-1を返します。

M_USE

MUSIC3

書　　式	m_use([t])
引　　数	char
戻り値	int
機　　能	指定のトラックバッファtの使用容量を返します。

　　t・・・・・・・・・トラック番号(1～80)

tを省略した場合は、全てのトラックバッファの合計使用容量を返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_VEL

MUSIC3

書　　式	m_vel([c] [,vel])
引　　数	char
戻り値	int
機　　能	指定のチャンネルcのベロシティを設定します。

　　c・・・・・・・・・チャンネル番号(1～25)
　　vel・・・・・・・ベロシティデータ(0～127)

cが省略された場合は、全てのチャンネルに vel を設定します。
velが省略された場合は、現在の設定値を返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

M_VGET

MUSIC

書　式	m_vget(vo,ca)
引　数	char(vo),char型二次元(4,10)配列名(ca)
戻り値	int
機　能	音色データを配列caに読み込みます。

vo・・・・・・・・音色番号(1～200)
ca・・・・・・・・音色データを格納するためのchar型二次元配列名。
配列の次元は、(4,10)。

戻り値として、正常なときは0、エラーのときは-1を返します。

➡ m_vset

M_VOL

MUSIC3

書　式	m_vol([c] [,vol])
引　数	char
戻り値	int
機　能	指定のチャンネルcにボリュームvolを設定します。

c・・・・・・・・・チャンネル番号(1～25)
vol・・・・・・・ボリュームデータ(0～127)

cが省略された場合は、全てのチャンネルに vol を設定します。
volが省略された場合は、現在の設定値を返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_mstvol

M_VOLFLT

書　式	m_volflt([c] [,md])
引　数	char
戻り値	int
機　能	MMLデータのボリューム設定コマンドVn,@Vnを、指定チャンネルの演奏時に出力するかどうかを選択します。

c・・・・・・・・・チャンネル番号(1～25)
md・・・・・・・・0=Vn,@Vnを出力する
　　　　　　　　1=Vn,@Vnを出力しない

cが省略された場合は、全てのチャンネルに md を設定します。
mdが省略された場合は、現在の設定値を返します。
戻り値として、正常のときは0、エラーのときは-1を返します。

➡ m_panflt,m_pgnflt

M_VSET

書　式　m_vset(vo,ca)

引　数　char(vo),char型二次元(4,10)配列名(ca)

戻り値　int

機　能　配列caに格納されている音色データを指定された音色番号voのメモリバッファに登録します。

vo・・・・・・・・音色番号(1～200)。1～68は登録済。
ca・・・・・・・・音色データが格納されているchar型二次元配列名。
配列の次元は(4,10)。

音色データの配列形式

	0		1～4(オペレータ)	
0	フィードバック／アルゴリズム	(0～63)	AR	(0～31)
1	スロットマスク	(0～15)	D1R	(0～31)
2	ウェーブフォーム	(0～3)	D2R	(0～31)
3	シンクロ	(0,1)	RR	(0～15)
4	スピード	(0～255)	D1L	(0～15)
5	PMD	(0～127)	TL	(0～127)
6	AMD	(0～7)	KS	(0～3)
7	PMS	(0～3)	MUL	(0～15)
8	AMS	(0～3)	DT1	(0～7)
9	L，RPAN	(0～3)	DT2	(0～3)
10			AMSイネーブル	(0,1)

なお、オペレータ1はM1（モジュレータ1）、オペレータ2はC1（キャリア1）、オペレータ3はM2（モジュレータ2）、およびオペレータ4はC2（キャリア2）に対応しています。

戻り値として、正常のときは0、エラーのときは-1を返します。

➡　m_vget

M_YCOM

書　式	m_ycom(〔md〕)
引　数	char
戻り値	int
機　能	md=0でＹコマンドをＭＩＤＩのコントロールデータと解釈します。 md=1でＹコマンドをＯＰＭレジスタの書き換えと解釈します。 mdを省略した場合は現在のモードを返します。（デフォルト=1） 戻り値として、正常のときは0、エラーのときは-1を返します。

1.8 ライブラリ一覧

● ライブラリ（ＯＰＭコール番号順）

番号	名　前	機　能	頁
$00	m_init	トラックバッファの初期化	55
$01	m_alloc	指定トラックのトラックバッファの確保	48
$02	m_assign	指定チャンネルのトラック設定	49
$03	m_vget	ＦＭ音源の音色データ取得	86
$04	m_vset	ＦＭ音源の音色を設定	88
$05	m_tempo	テンポ指定	82
$06	m_trk	指定トラックにＭＭＬデータをセット	83
$07	m_free	指定トラックバッファの残りバイト数を返す	54
$08	m_play	指定チャンネルの演奏開始	75
$09	m_stat	指定チャンネルの演奏状態を返す	78
$0A	m_stop	指定チャンネルの演奏を一時停止	79
$0B	m_cont	指定チャンネルの演奏再開	51
$0C	m_atoi	指定トラックバッファの先頭アドレスを返す	49
$0F	m_use	指定トラックバッファの使用容量を返す	84
$10	m_chan	データ出力チャンネルを一時変更	50
$11	m_prog	指定チャンネルの音色番号設定	76
$12	m_vol	指定チャンネルにボリューム設定	86
$13	m_pan	指定チャンネルのパンポット指定	67
$14	m_vel	指定チャンネルのベロシティ設定	85
$15	m_trns	指定チャンネルの音程シフト	84
$18	m_antoff	指定チャンネルへのオールノートオフ出力	48
$19	m_dirout	ＭＩＤＩ出力	52
$1A	m_enable	指定チャンネルの発音許可	52
$1B	m_nton	指定チャンネルへノートオン出力	64
$1C	m_ntoff	指定チャンネルへノートオフ出力	63
$20	m_sync	ＭＩＤＩの同期モードを設定	80
$21	m_meter	曲の拍子設定	57
$29	m_sndset	SNDﾌｧｲﾙのデータをＦＭ音源用に登録	77
$2A	m_pcmset	指定の音程にＰＣＭ音を登録	73
$2B	m_sysch	チャンネル番号の割り当て変更	81
$2C	m_ifchk	ＭＩＤＩインターフェイスの有無チェック	55
$2F	m_freea	ﾄﾞﾗｲﾊﾞ起動時のﾄﾗｯｸﾊﾞｯﾌｧの全ﾊﾞｲﾄ数を返す	54
$30	m_out	1バイトデータをＭＩＤＩへ出力	67
$32	m_meas	演奏中の小節番号を返す	57
$33	m_ycom	Ｙコマンドの解釈	89
$34	m_start	演奏開始小節の設定	78
$35	m_end	演奏停止小節の設定	53
$36	m_modsns	ＦＭ音源のモジュレーションの深さ設定	58
$38	m_pcmrec	外部入力音のＰＣＭ録音	72
$39	m_pcmget	登録されているＰＣＭデータの取得	70
$3A	m_pcmlen	指定されたＰＣＭデータのサイズを返す	70
$3B	m_pcmbsy	ＰＣＭの実行状態を返す	69
$3C	m_pcmon	指定されたＰＣＭデータの再生	71
$3D	m_ctrlres	指定チャンネルへの各種のオフ出力	51
$3E	m_opmreg	指定のＯＰＭレジスタにデータをセット	66
$3F	m_mdreg	ＭＩＤＩ制御ＩＣのレジスタにデータをセット	56
$40	m_tnmset	ＦＭ音源の音色名の登録	83
$41	m_tnmget	ＦＭ音源の音色名の取得	82
$42	m_pnmset	ＰＣＭ音色名の登録	76

番号	名　　前	機　　　能	頁
$43	m_pnmget	ＰＣＭ音色名の取得	75
$44	m_mstvol	マスターボリュームの設定	61
$45	m_bend	指定チャンネルのピッチベンド設定	50
$46	m_mod	指定チャンネルのモジュレーション設定	58
$47	m_opmexc	ＦＭ音源のパラメータセットの選択	65
$48	m_opmlfq	ＦＭ音源のＬＦＯの発振周波数の設定	65
$49	m_pgmflt	ＭＭＬ音色切り替えコマンドの出力選択	74
$4A	m_volflt	ＭＭＬボリューム設定コマンドの出力選択	87
$4B	m_panflt	ＭＭＬパンポット設定コマンドの出力選択	68
$5E	m_version	OPMDRV3.Xのバージョンと作成日付を返す	85
	m_errget	直前のMUSIC関数のエラーコードを返す	53
	m_mcont	ビットパターン指定チャンネルの演奏再開	56
	m_mplay	ビットパターン指定チャンネルの演奏開始	59
	m_mstop	ビットパターン指定チャンネルの演奏一時停止	60
	m_mute	指定チャンネルの音を消音	62
	m_pcmclr	ＰＣＭバッファのクリア	69
	m_solo	指定チャンネル以外の音を消音	77

1.9 ライブラリ リファレンス

「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ　ｖｅｒ２．１」では、（ＦＭ音源／ＭＩＤＩ／ＡＤＰＣＭを制御できる）ＯＰＭＤＲＶ３．Ｘ用のライブラリがサポートされています。

これにより、新しく次のファイルがｖｅｒ２．１より追加／変更されています。

ファイル名	内　　容
MUSIC.H	FM音源（MIDI/ADPCM対応）用関数の定義
MUSIC3.H	FM音源/MIDI/ADPCM用拡張関数の定義
BASLIB.L	ＢＡＳＴＯＣライブラリ

なお、「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ　ｖｅｒ２．０」でサポートされていたＯＰＭＤＲＶ２．Ｘ用のライブラリ（「Ｃコンパイラライブラリーマニュアル１」を参照）は新しいＯＰＭＤＲＶ３．Ｘ用のライブラリではサポートしていませんのでご注意ください。

従来のＯＰＭＤＲＶ２．Ｘ用のライブラリをご使用になりたい場合は、システムディスク２のＭＵＳＩＣディレクトリに入っているファイルをご使用ください。
（ＯＰＭＤＲＶ３．Ｘ用のライブラリと同時に利用することはできません）

m_alloc

レベル3

書　式

```
#include <music.h>

int m_alloc(t,s);
char t;        /*  トラック番号(1～80)  */
int s;         /*  バッファサイズ(1～65535)  */
```

機　能

指定トラックのトラックバッファを確保します。
全トラックバッファ内のデータはクリアされます。
また、バッファサイズに1～3を指定しても、4バイト確保します。
tをに-1または'NASI'を指定した場合、全てのトラックをバッファサイズsで確保します。
なお、総トラックのバッファサイズ（それぞれのトラックに割り当てられているトラックバッファの総合計）は、ＣＯＮＦＩＧ．ＳＹＳファイル内のＯＰＭＤＲＶ３．Ｘのあとに指定して、再度システム起動することにより変更できます。

```
DEVICE = ¥SYS¥OPMDRV3.X  #nnn
```

（nnn：総トラックバッファサイズ［単位：ＫＢ］）

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_antoff

レベル3

書　式

```
#include <music3.h>

int m_antoff(c);
int c;         /*  チャンネル番号（1～25）  */
```

機　能

指定のチャンネルへオールノートオフを出力します。
cに-1または'NASI'を指定した場合、全ての チャンネルへオールノートオフを出力します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_assign

レベル3

書式

```
#include <music.h>

int m_assign(c,t);
char c;       /* チャンネル番号(1～25) */
char t;       /* トラック番号(1～80) */
```

機能

指定のチャンネルcで使用するトラックtを設定します。
tに-1または'NASI'を指定した場合、チャンネルcに設定されているトラック番号を返します。
複数のチャンネルに同一のトラックを割りあてると、それぞれ同じ楽譜データを演奏します。
また、複数のトラックを１つのチャンネルに割りあてると、最も新しく割りふられたトラックの楽譜データを演奏します。
デフォルト値は、チャンネル１～２５が、それぞれトラック１～２５に対応します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
tに-1または'NASI'を指定した場合、チャンネルcに設定されているトラック番号を返します。

m_atoi

レベル3

書式

```
#include <music.h>

int m_atoi(t);
char t;       /* トラック番号 (1～80) */
```

機能

トラックtのトラックバッファ先頭アドレスを返します。
トラックバッファは、ＯＰＭＤＲＶ３．Ｘの内部バッファのため、スーパーバイザ領域です。
データのアクセスには、注意してください。

戻り値

トラックバッファの先頭アドレスを返します。

m_bend

レベル3

書式

```
#include <music3.h>

int m_bend(c,b);
int c;          /*  チャンネル番号(1~25)  */
char b;         /*  ピッチベンドデータ(0~127)  */
```

機能

指定のチャンネルのピッチベンドを設定します。
cに-1または'NASI'を指定した場合、全てのチャンネルに b を設定します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_chan

レベル3

書式

```
#include <music3.h>

int m_chan(c1,c2);
char c1;        /*  チャンネル番号(1~25)  */
int c2;         /*  出力チャンネル番号(1~25)  */
```

機能

指定のチャンネルのデータ出力チャンネルを一時的に変更します。
c2に-1または'NASI'を指定した場合、現在の設定値を返します。
m_assign関数やＭＭＬデータ @N とは異なり、チャンネル変更前のパラメータ(トラック番号、音色番号、ボリュームなど)を引き継ぎます。
m_init関数実行まで有効です。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_init

m_cont

レベル3

書　式

```
#include <music.h>

int m_cont(c1,c2,c3,c4,c5,c6,c7,c8);
int c1,c2,c3,c4,c5,c6,c7,c8;     /*  チャンネル番号 (1～25) */
```

機　能

c1からc8で指定されたチャンネルにおいて、一時停止した演奏を再開します。
指定するチャンネルの個数が７個以下の場合は余りの引数に'NASI'を指定しなければなりません。
c1～c8が全て'NASI'ならば、1～25チャンネル中、現在一時停止していたチャンネルを全て演奏再開します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_stop

m_ctrlres

レベル3

書　式

```
#include <music3.h>

int m_ctrlres(c);
int c;       /*  チャンネル番号(1～25)  */
```

機　能

指定のチャンネルへピッチベンドオフ／モジュレーションオフ／ダンパーオフ／オールノートオフを出力します。
cに-1または'NASI'を指定した場合、全てのチャンネルへピッチベンドオフ／モジュレーションオフ／ダンパーオフ／オールノートオフを出力します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_dirout

レベル3

書　式

```
#include <music3.h>

int m_dirout(lng,ca);
int lng;        /*  データサイズ  */
char *ca;       /*  ＭＩＤＩデータの先頭アドレス  */
```

機　能

lngバイトのＭＩＤＩデータをＭＩＤＩへ出力します。

戻り値

正常終了の場合は0を返します。エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_out

m_enable

レベル3

書　式

```
#include <music3.h>

int m_enable(cp);
int cp;         /*  チャンネルのビットパターン  */
```

機　能

ビットパターンで指定されたチャンネルの発音を許可します。
ビット0～24がチャンネル1～25に対応しており、発音を許可したいチャンネルに対応したビットに１を指定します（他のビットは常に0です）。
cpに0を指定した場合、全てのチャンネルの発音を許可します。
cpに-1または‘NASI’を指定した場合、全てのチャンネルの状態をbit0～24に返します。
m_mute関数とm_solo関数のミュート設定は解除されます。m_mstop関数とは異なり、ノート以外のデータは常に出力されています（ミュート状態）。

戻り値

正常終了の場合は0を返します。エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_mcont,m_mplay,m_mstop,m_mute,m_solo

m_end

レベル3

書　式

```
#include <music3.h>

int m_end(m);
int m;          /*  小節番号  */
```

機　能

演奏を停止する小節を設定します。
全てのチャンネルの|:　D.S　D.C　:|などの位置が同じでないと正常な演奏ができません。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_start

m_errget

レベル3

書　式

```
#include <music3.h>

int m_errget();
```

機　能

直前のＭＵＳＩＣ各関数(m_で始まる関数)のエラーコードを返します。
エラーコードについては、「１．１３エラーコード一覧」を参照してください。

戻り値

正常終了している場合は0を返し、エラーの場合は1～65のエラーコードを返します。

m_free

レベル3

書　式

```
#include <music.h>

int m_free(t);
char t;        /*  トラック番号(1～80)  */
```

機　能

指定のトラックtのトラックバッファの残りバイト数を返します。

戻り値

トラックバッファの残りバイト数を返します。

参照関数

m_freea

m_freea

レベル3

書　式

```
#include <music3.h>

int m_freea();
```

機　能

ドライバ起動時に確保したトラックバッファの全バイト数を返します。

戻り値

トラックバッファの全バイト数を返します。

参照関数

m_free,m_use

m_ifchk

レベル3

書　式

```
#include <music3.h>

int m_ifchk();
```

機　能

ＭＩＤＩインターフェイスの有無をチェックします。

戻り値

インターフェイスがつながっている場合は 0、つながっていない場合は -1を返します。

m_init

レベル3

書　式

```
#include <music.h>

int m_init(md);
int md;       /*  初期化モード  */
```

機　能

トラックバッファを初期化します。
各トラックバッファは４バイトに設定されます。
また、各チャンネルのパラメータ（トラック番号、音色番号、ボリュームなど）も初期化されます。
mdに1を指定した場合、ＦＭ音源の音色データも初期化します。
mdに0または'NASI'を指定した場合は音色データは初期化しません。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_mcont

レベル3

書　式

```
#include <music3.h>

int m_mcont(cp);
int cp;      /*  チャンネルのビットパターン  */
```

機　能

ビットパターンで指定されたチャンネルの演奏を再開します。
ビット0～24がチャンネル1～25に対応しており、演奏を再開したいチャンネルに対応したビットに１を指定します（他のビットは常に0です）。
cpに0または-1、'NASI'を指定した場合、全てのチャンネルの演奏を再開します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_enable,m_mplay,m_mstop

m_mdreg

レベル3

書　式

```
#include <music3.h>

int m_mdreg(gr,adr,dt);
char gr;      /*  グループ番号  */
char adr;     /*  アドレス番号  */
char dt;      /*  書き込むデータ  */
```

機　能

ＭＩＤＩ制御ＩＣ（ＹＭ３８０２）のレジスタに値を書き込みます。
grにはグループ番号、adrにはアドレス番号、dtにはレジスタに書き込む値を0～255の範囲で指定します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_meas

レベル3

書　式

```
#include <music3.h>

int m_meas();
```

機　能

現在演奏中の小節番号を返します。
|: D.S. などのくり返し記号の場合はくり返し回数がそのまま加算されますので、実際の小節番号とは異なる場合があります。

戻り値

現在演奏中の小節番号

参照関数

m_end,m_start

m_meter

レベル3

書　式

```
#include <music3.h>

int m_meter(n,d);
char n;  /*  分子(2～16)  */
char d;  /*  分母(2,4,8,16)  */
```

機　能

曲の拍子を設定します。
デフォルト値は4/4拍子に設定されています。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_mod

レベル3

書式

```
#include <music3.h>

int m_mod(c,m);
int c;  /*  チャンネル番号(1〜25)  */
int m;  /*  モジュレーションデータ(0〜127)  */
```

機能

指定のチャンネルのモジュレーションを設定します。
cに-1または'NASI'を指定した場合は、全てのチャンネルに m を設定します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_modsns

m_modsns

レベル3

書式

```
#include <music3.h>

int m_modsns(m);
int m;  /*  モジュレーションの深さ(0〜127)  */
```

機能

FM音源全体のモジュレーションの深さを設定します。
mに-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
mに-1または'NASI'を指定した場合は、現在の設定値を返します。

参照関数

m_mod

m_mplay

レベル3

書　式

```
#include <music3.h>

int m_mplay(cp);
int cp;  /*  チャンネルのビットパターン  */
```

機　能

ビットパターンで指定されたチャンネルの演奏を開始します。
ビット0～24がチャンネル1～25に対応しており、演奏を開始したいチャンネルに対応したビットに１を指定します（他のビットは常に0です）。
cpに0または-1、'NASI'を指定した場合、全てのチャンネルの演奏を開始します。

例）チャンネル1～13とチャンネル22～25の演奏を開始します。

```
m_mplay(&b1111000000001111111111111)
```

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_enable,m_mcont,m_mstop

m_mstop

レベル3

書　式

```
#include <music3.h>

int m_mstop(cp);
int cp;  /*  チャンネルのビットパターン  */
```

機　能

ビットパターンで指定されたチャンネルの演奏を一時停止します。
ビット0～24がチャンネル1～25に対応しており、演奏を一時停止したいチャンネルに対応したビットに１を指定します（他のビットは常に0です）。
cpに0または-1、'NASI'を指定した場合、全てのチャンネルの演奏を一時停止します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_enable,m_mcont,m_mplay

m_mstvol

レベル3

書　式

```
#include <music3.h>

int m_mstvol(v);
int v;  /*  ボリュームデータ(0～127) */
```

機　能

マスターボリュームを設定します。
デフォルト値は127に設定されています。
vに-1または'NASI'を指定した場合、現在の設定値を返します。
m_init関数を実行するまで有効です。
ＰＣＭのチャンネルに対しては無効です。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
vに-1または'NASI'を指定した場合、現在の設定値を返します。

参照関数

m_init,m_vol

m_mute

レベル３

書　式

```
#include <music3.h>

int m_mute(c1,c2,c3,c4,c5,c6,c7,c8);
int c1,c2,c3,c4,c5,c6,c7,c8; /*  チャンネル番号(1～125)   */
```

機　能

c1からc8で指定したチャンネルの音をミュート(消音)します。
ノート以外のデータは常に出力されます。
c1からc8が全て 'NASI' ならば全てのチャンネルのミュートが解除されます。
8個以上のチャンネルを同時にミュートする場合は、m_enable関数を使用してください。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_enable, m_solo

m_ntoff

レベル3

書　式

```
#include <music3.h>

int m_ntoff(c,nt,vel);
char c;    /*  チャンネル番号(1～25)  */
char nt;   /*  ノート番号(0～127)  */
int vel;   /*  ベロシティ(0～127)  */
```

機　能

指定のチャンネルへノートオフを出力します。
velに-1または'NASI'を指定した場合、ベロシティ値を６４で出力します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_nton

m_nton

レベル３

書　式

```
#include <music3.h>

int m_nton(c,nt,vel);
char c;    /*  チャンネル番号(1～25)  */
char nt;   /*  ノート番号(0～127)  */
int vel;   /*  ベロシティ(0～127)  */
```

機　能

指定のチャンネルへノートオンを出力します。
velに-1または‘NASI’を指定した場合、ベロシティ値を６４で出力します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_ntoff

m_opmexc

レベル3

書　式

```
#include <music3.h>

int m_opmexc(md);
int md;  /*  パラメータなどの設定スイッチ  */
```

機　能

ＦＭ音源にセットするパラメータの内、ＬＦＱ／ＰＭＤ／ＡＭＤなどの全体に影響するパラメータを音色切り替え時にセットするかどうかを選択します。
0を指定するとセットしません。
1を指定するとセットします。デフォルト値は1です。
mdに-1または'NASI'を指定した場合、現在の状態を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
mdに-1または'NASI'を指定した場合、現在の状態を返します。

参照関数

m_vset

m_opmlfq

レベル3

書　式

```
#include <music3.h>

int m_opmlfq(spd);
int spd;  /*  スピード  */
```

機　能

ＦＭ音源のＬＦＯの発振周波数（スピード）を設定します。
0～255を指定します。デフォルト値は198です。
spdに-1または'NASI'を指定した場合、現在のデータを返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
spdに-1または'NASI'を指定した場合、現在のデータを返します。

m_opmreg

レベル3

書　式

```
#include <music3.h>

int m_opmreg(reg,data);
int reg;     /* ＯＰＭレジスタ番号(0～255) */
int data;    /* ＯＰＭへ書き込むデータ（0～255）*/
```

機　能

指定のＯＰＭレジスタにデータをセットします。
dataに-1または'NASI'を指定した場合は、現在のデータを返します。
一度も指定のＯＰＭレジスタが書き換えられていない場合は、－１を返します。
reg,data両方に-1または'NASI'を指定した場合は、ＯＰＭレジスタバッファのポインタを返します。
ＯＰＭレジスタバッファは、各レジスタにつき２バイトずつ計５１２バイトあり、現在までにレジスタが書き換えられなかった場合は、そのレジスタ位置に－１（ワード）がセットされています。
ＯＰＭレジスタバッファは、ＯＰＭＤＲＶ３．Ｘの内部バッファのため、スーパーバイザー領域です。
データのアクセスには、注意してください。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
dataに-1または'NASI'を指定した場合は、現在のデータを返します。
reg,data両方に-1または'NASI'を指定した場合は、ＯＰＭレジスタバッファのポインタを返します。

m_out

レベル3

書　式

```
#include <music3.h>

int m_out(data);
char data;    /* ＭＩＤＩデータ */
```

機　能

１バイトデータをＭＩＤＩへ出力します。

戻り値

正常終了の場合は0を返します。エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_dirout

m_pan

レベル3

書　式

```
#include <music3.h>

int m_pan(c,pan);
int c;     /* チャンネル番号(1～25) */
int pan;   /* パンポットデータ(0～127) */
```

機　能

指定のチャンネルのパンポットを設定します。
パンポットデータ

0　：ライト（右端）
⋮
64　：センター（中央）
⋮
127：レフト（左端）

c に -1または'NASI'を指定した場合は、全てのチャンネルに pan を設定します。
pan に-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
panに-1または'NASI'を指定した場合は、現在の設定値を返します。

m_panflt

レベル3

書　式

```
#include <music3.h>

int m_panflt(c,md);
int c;  /* チャンネル番号(1～25) */
int md; /* フィルタモード */
```

機　能

ＭＭＬデータ内のＰＡＮの切り替えコマンドを、指定チャンネルの演奏時に出力するかどうかを選択します。
mdに0を指定した場合、フィルタをオフします。
mdに1を指定した場合、フィルタをオンします。
cに-1または'NASI'を指定した場合は、全てのチャンネルに md を設定します。
mdに-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
mdに-1または'NASI'を指定した場合は、現在の設定値を返します。

参照関数

m_pgmflt,m_volflt

m_pcmbsy

レベル3

書式

```
#include <music3.h>

int m_pcmbsy();
```

機能

ＰＣＭの実行状態を調べます。

戻り値

ＰＣＭの実行状態を以下に示す値で返します。

0x00 ： 何も実行していない
0x02 ： 出力中（ADPCMOUT,m_pcmon,a_playを実行中）
0x04 ： 入力中（ADPCMINP,m_pcmrec,a_recを実行中）
0x12 ： 出力中（ADPCMAOTを実行中）
0x14 ： 入力中（ADPCMAINを実行中）
0x22 ： 出力中（ADPCMLOTを実行中）
0x24 ： 入力中（ADPCMLINを実行中）

m_pcmclr

レベル3

書式

```
#include <music3.h>

int m_pcmclr();
```

機能

ＰＣＭバッファをクリアします。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_pcmget

レベル3

書　式

```
#include <music3.h>

int m_pcmget(nt,ca,s);
char nt;  /* ノート番号 (0～127) */
char *ca; /* ＰＣＭデータバッファの先頭アドレス */
int s;    /* バッファサイズ */
```

機　能

登録されているＰＣＭデータをcaにバッファサイズ分読み込みます。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_pcmlen,m_pcmset

m_pcmlen

レベル3

書　式

```
#include <music3.h>

int m_pcmlen(nt);
int nt; /* ノート番号 (0～127) */
```

機　能

指定されたＰＣＭデータのサイズを返します。
ntに-1または'NASI'を指定した場合は、ＰＣＭバッファの残り容量を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
ntに-1または'NASI'を指定した場合は、ＰＣＭバッファの残り容量を返します。

参照関数

m_pcmset,m_pcmget

m_pcmon

レベル3

書　式

```
#include <music3.h>

int m_pcmon(nt,sf,md);
char nt;  /* ノート番号(0～127) */
int sf;   /* サンプリング周波数(0～4) */
int md;   /* 音声出力モード(1～3) */
```

機　能

指定されたＰＣＭデータを再生します。
sfとmd両方に-1または'NASI'を指定した場合は、サンプリング周波数=4 音声出力モード=3で再生します。

サンプリング周波数
　0・・・・・・・・・・・・・ 3.9KHz
　1・・・・・・・・・・・・・ 5.2KHz
　2・・・・・・・・・・・・・ 7.8KHz
　3・・・・・・・・・・・・・10.4KHz
　4・・・・・・・・・・・・・15.6KHz

音声出力モード（オーディオ出力端子）
　1・・・・・・・・・・・・・左
　2・・・・・・・・・・・・・右
　3・・・・・・・・・・・・・ステレオ（左右両方）

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_pcmrec,m_pcmset

m_pcmrec

レベル3

書　式

```
#include <music3.h>

int m_pcmrec(sf,lng,ca);
char sf;  /* サンプリング周波数(0～4) */
int lng;  /* データサイズ(最大=32767) */
char *ca; /* ＰＣＭデータバッファの先頭アドレス */
```

機　能

外部入力音（オーディオ入力端子）をＰＣＭ録音します。
実行直後に入力待ちになり、約４秒以内に入力がなければ実行を中止します。

サンプリング周波数

0･････････････ 3.9KHz（一秒間に1950ﾊﾞｲﾄ）
1･････････････ 5.2KHz（一秒間に2600ﾊﾞｲﾄ）
2･････････････ 7.8KHz（一秒間に3900ﾊﾞｲﾄ）
3･････････････10.4KHz（一秒間に5200ﾊﾞｲﾄ）
4･････････････15.6KHz（一秒間に7800ﾊﾞｲﾄ）

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_pcmon,m_pcmset

m_pcmset

レベル3

書　式

```
#include <music3.h>

int m_pcmset(nt,sf,lng,ca);
char nt;  /* ノート番号(0～127) */
char sf;  /* サンプリング周波数(0～4) */
int lng;  /* データサイズ(最大=32767) */
char *ca; /* ＰＣＭデータの先頭アドレス */
```

機　能

指定の音程にＰＣＭ音を登録します。
指定できる音程は'C-2'(0)から'G#8'(127)までです。

サンプリング周波数

0・・・・・・・・・・・・・ 3.9KHz (一秒間に1950ﾊﾞｲﾄ)
1・・・・・・・・・・・・・ 5.2KHz (一秒間に2600ﾊﾞｲﾄ)
2・・・・・・・・・・・・・ 7.8KHz (一秒間に3900ﾊﾞｲﾄ)
3・・・・・・・・・・・・・10.4KHz (一秒間に5200ﾊﾞｲﾄ)
4・・・・・・・・・・・・・15.6KHz (一秒間に7800ﾊﾞｲﾄ)

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_pcmget, m_pcmlen, m_pcmon, m_pcmrec

m_pgmflt

レベル3

書　式

```
#include <music3.h>

int m_pgmflt(c,md);
int c;  /* チャンネル番号(1～25) */
int md; /* フィルタモード */
```

機　能

MMLデータ内の音色の切り替えコマンドを、指定チャンネルの演奏時に出力するかどうかを選択します。
mdに0を指定した場合、フィルタをオフします。
mdに1を指定した場合、フィルタをオンします。
cに-1または'NASI'を指定した場合は、全てのチャンネルに md を設定します。
mdに-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
mdに-1または'NASI'を指定した場合は、現在の設定値を返します。

参照関数

m_panflt,m_volflt

m_play

レベル3

書　　式

```
#include <music.h>

int m_play(c1,c2,c3,c4,c5,c6,c7,c8);
int c1,c2,c3,c4,c5,c6,c7,c8;  /* チャンネル番号(1～25) */
```

機　　能

ＦＭ音源の指定したチャンネルに割りあてたトラックデータを演奏開始します。
指定するチャンネルの個数が７個以下の場合は、余りの引数に 'NASI' を指定しなければなりません。
c1～c8が全て'NASI'ならば1～25チャンネルを全て演奏開始します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_pnmget

レベル3

書　　式

```
#include <music3.h>

char *m_pnmget(nt);
char nt;  /* ノート番号(0～127) */
```

機　　能

ＰＣＭのノート番号に登録されている音色名を取得します。
m_pnmget関数とm_tnmget関数は、どちらでも続けて使用する場合、必ず戻り値のポインタが示す文字列をコピーしてから使用してください。

戻り値

ＰＣＭの音色名を示す文字列のポインタを返します。

参照関数

m_pnmset

m_pnmset

レベル3

書　式

```
#include <music3.h>

int m_pnmset(nt,pn);
char nt    /* ノート番号(0～127) */
char *pn   /* PCMの音色名の先頭のアドレス */
```

機　能

ＰＣＭのノート番号に音色名を登録します。(半角10文字以内)

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_pnmget

m_prog

レベル3

書　式

m_prog(c,p)

```
int m_prog(c,p);
int c;   /* チャンネル番号(1～25) */
int p;   /* 音色番号(ＦＭ音源:1～200、ＭＩＤＩ:1～128) */
```

機　能

指定のチャンネルの音色番号を設定します。
c に-1または'NASI'を指定した場合は、全てのチャンネルに p を設定します。
p に-1または'NASI'を指定した場合は、現在の設定値を返します。
データサイズは、１音色80バイトで最大200音色分（＝16000バイト）まで指定できます。

戻り値

正常終了の場合は0を返します。

エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
p に-1または'NASI'を指定した場合は、現在の設定値を返します。

m_sndset

レベル3

書式

```
#include <music3.h>

int m_sndset(ca,s);
char *ca;  /* ＳＮＤファイル形式のデータ */
int s;     /* データサイズ（最大=16000） */
```

機能

SOUND PRO-68K のＳＮＤファイル形式のデータをＦＭ音源用の音色データとして登録します。
同時に音色名も登録します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_solo

レベル3

書式

```
#include <music3.h>

int m_solo(c);
int c;  /* チャンネル番号(1～25) */
```

機能

指定されたチャンネル以外の全ての音をミュート（消音）します。
ノート以外のデータは、常に出力されます。
cに0または-1、'NASI'を指定した場合、解除されます。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
cに0または-1、'NASI'を指定した場合、解除されます。

参照関数

m_enable,m_mute

m_start

レベル3

書式

```
#include <music3.h>

int m_start(m);
int m;  /* 演奏を開始する小節番号 */
```

機能

演奏を開始する小節を設定します。（演奏が終わるとクリアされる）
全てのチャンネルの |: D.S. D.C. :| などの数や位置が同じでないと正常な演奏ができません。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_end, m_meas, m_play

m_stat

レベル3

書式

```
#include <music.h>

int m_stat(c);
int c;  /* チャンネル番号(1～25) */
```

機能

指定のチャンネルの演奏状態を調べます。
cに-1または'NASI'を指定した場合は、全てのチャンネルの状態をbit0～24に返します。
bit0～24がチャンネル1～25に対応しています。

戻り値

演奏中なら1を、演奏終了なら0を返します。
cに-1または'NASI'を指定した場合は、全てのチャンネルの状態をbit0～24に返します。

m_stop

レベル3

書　式

```
#include <music.h>

int m_stop(c1,c2,c3,c4,c5,c6,c7,c8);
int c1,c2,c3,c4,c5,c6,c7,c8;  /* チャンネル番号(0～25) */
```

機　能

指定のチャンネルの演奏を一時停止します。
指定するチャンネルの個数が７個以下の場合は、余りの引数に'NASI'を指定しなければなりません。
c1～c8が全て'NASI'ならば、1～25チャンネル中、現在演奏していたチャンネルすべてを一時停止します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_sync

レベル３

書　式

```
#include <music3.h>

int m_sync(c);
int c;          /* 同期モード */
```

機　能

ＭＩＤＩの同期モードを設定します。

同期モード
　　0・・・・・・・・・内部のＭＩＤＩクロックに合わせて演奏
　　1・・・・・・・・・外部のＭＩＤＩクロックに合わせて演奏
　　2・・・・・・・・・外部のＦＳＫクロックに合わせて演奏

外部同期にする場合は、m_sync(1) 実行後 m_play関数で待機状態になりＭＩＤＩのスタートコマンド($FA)を待ちます。
cに-1または'NASI'を指定した場合は、現在の状態を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
cに-1または'NASI'を指定した場合は、現在の状態を返します。

m_sysch

レベル3

書　式

```
#include <music3.h>

int m_sysch(cm);
char *cm;  /* 割り当てモードを示す文字列の先頭アドレス */
```

機　能

各音源に対するチャンネル番号の割り当てを変更します。
(ドライバ組み込み時のオプションスイッチと同じ)
cmに-1または'NASI'を指定した場合は、現在のモードを数値で返します。

割り当てモード

0="OPM"・・・・・・・OPM=ch1～8	PCM=ch9	MIDI=ch10～25
1="OMP"・・・・・・・OPM=ch1～8	MIDI=ch9～24	PCM=ch25
2="MOP"・・・・・・・MIDI=ch1～16	OPM=ch17～24	PCM=ch25
3="MPO"・・・・・・・MIDI=ch1～16	PCM=ch17	OPM=ch18～25
4="POM"・・・・・・・PCM=ch1	OPM=ch2～9	MIDI=ch10～25
5="PMO"・・・・・・・PCM=ch1	MIDI=ch2～17	OPM=ch18～25

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
cmに-1または'NASI'を指定した場合は、現在のモードを数値で返します。

m_tempo

レベル3

書式

```
#include <music.h>

int m_tempo(te);
int te;  /* テンポ */
```

機能

ＦＭ音源およびＭＩＤＩの演奏時のテンポを指定します。
テンポteを１分間に４分音符を何回打つかで指定します。
テンポは、20〜300の範囲です。（デフォルト値は120）
teに-1または'NASI'を指定した場合は、現在のテンポを返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
teに-1または'NASI'を指定した場合は、現在のテンポを返します。

m_tnmget

レベル3

書式

```
#include <music3.h>

char *m_tnmget(vo);
char vo;  /* 音色番号(1〜200) */
```

機能

ＦＭ音源の音色番号で示される音色データに登録されている音色名を取得します。
m_pnmget関数とm_tnmget関数は、どちらでも続けて使用する場合、必ず戻り値のポインタが示す文字列をコピーしてから使用してください。

戻り値

音色名を示す文字列のポインタを返します。

参照関数

m_tnmset

m_tnmset

レベル3

書　式

```
m_tnmset(vo,tn)

int m_tnmset(vo,tn);
char vo;    /* 音色番号(1～200) */
char *tn; /* 音色名を示す文字列の先頭アドレス */
```

機　能

ＦＭ音源の音色番号で示される音色データに音色名を登録します。
（半角10文字以内）

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_tnmget

m_trk

レベル3

書　式

```
#include <music.h>

int m_trk(t,st);
char t;     /* トラック番号(1～80) */
char *st;   /* ＭＭＬデータ */
```

機　能

ＭＭＬによる楽譜データをトラックtに書き込みます。
厳密にいうとトラックtに楽譜データを追加していきます。
ＭＭＬデータの終端は、ヌル文字（0）です。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

m_trns

レベル3

書　式

```
#include <music3.h>

int m_trns(c,stp);
int c;    /* チャンネル番号 (1～25) */
int stp; /* 音程を移調するデータ(0～48) */
```

機　能

指定のチャンネルの音程をシフトします。（半音ステップ: ±2oct)
デフォルト値は24です。
c に-1または'NASI'を指定した場合は、全てのチャンネルに stp を設定します。
stp に-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
stp に-1または'NASI'を指定した場合は、現在の設定値を返します。

m_use

レベル3

書　式

```
#incluse <music3.h>

int m_use(t);
int t;   /* トラック番号(1～80) */
```

機　能

指定のトラックバッファの使用容量を返します。
t に-1または'NASI'を指定した場合は、全てのトラックバッファの合計使用容量を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
t に-1または'NASI'を指定した場合は、全てのトラックバッファの合計使用容量を返します。

m_vel

レベル3

書　式

```
#include <music3.h>

int m_vel(c,vel);
int c;    /* チャンネル番号(1～25) */
int vel; /* ベロシティデータ(0～127) */
```

機　能

指定のチャンネルのベロシティを設定します。
c に-1または'NASI'を指定した場合は、全てのチャンネルに vel を設定します。
vel に-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
vel に-1または'NASI'を指定した場合は、現在の設定値を返します。

m_version

レベル3

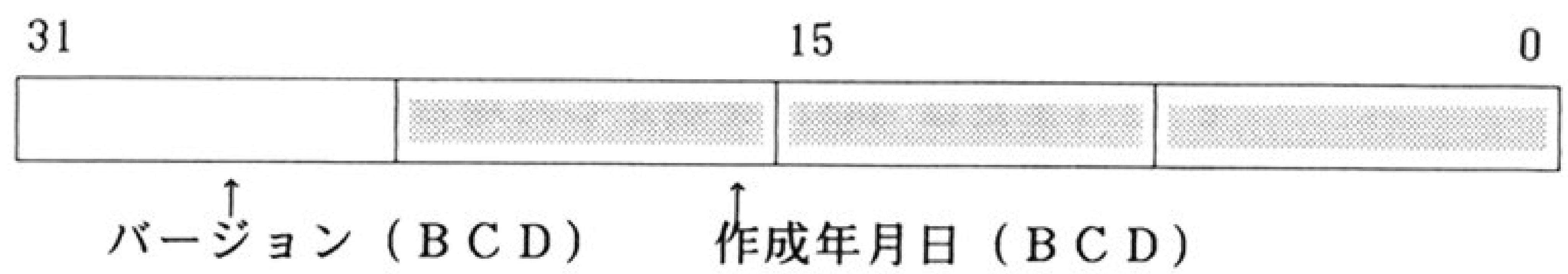
書　式

```
#include <music3.h>

int m_version ();
```

機　能

OPMDRV3.Xのバージョンと作成日付を返します。



例）0x10920303
　　バージョン1.0 1992年3月3日作成

戻り値

バージョンと作成日付を返します。

m_vget

レベル3

書　式

```
#include <music.h>

int m_vget(vo,ca);
char vo;  /* 音色番号 (1～200) */
char *ca; /* 音色データを格納するﾊﾞｯﾌｧの先頭ｱﾄﾞﾚｽ (55バイト) */
```

機　能

ＦＭ音源の音色データを取得します。
音色データをcaに読み込みます。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_vset

m_vol

レベル3

書　式

```
#include <music3.h>

int m_vol(c,vol);
int c;    /* チャンネル番号(1～25) */
int vol;  /* ボリュームデータ(0～127) */
```

機　能

指定のチャンネルcにボリュームvolを設定します。
c に-1または'NASI'を指定した場合は、全てのチャンネルに vol を設定します。
vol に-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
vol に-1または'NASI'を指定した場合は、現在の設定値を返します。

参照関数

m_mstvol

m_volflt

レベル3

書　式

```
#include <music3.h>

int m_volflt(c,md);
int c;  /* チャンネル番号(1～25) */
int md; /* フィルタモード */
```

機　能

ＭＭＬデータ内のボリューム設定コマンドを、指定チャンネルの演奏時に出力するかどうかを選択します。
mdに0を指定した場合、フィルタをオフします。
mdに1を指定した場合、フィルタをオンします。
c に-1または'NASI'を指定した場合は、全てのチャンネルにmdを設定します。
md に-1または'NASI'を指定した場合は、現在の設定値を返します。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
md に-1または'NASI'を指定した場合は、現在の設定値を返します。

参照関数

m_panflt,m_pgmflt

m_vset

レベル3

書　式

```
#include <music.h>

int m_vset(vo,ca);
char vo;  /* 音色番号(1～200) */
char *ca; /* 音色データの先頭アドレス（55バイト） */
```

機　能

ＦＭ音源の音色を設定します。
音色データのデータ形式

	0	1～4(ｵﾍﾟﾚｰﾀ)
0	フィードバック／アルゴリズム(0～63)	ＡＲ (0～31)
1	スロットマスク (0～15)	D1Ｒ (0～31)
2	ウェーブフォーム (0～3)	D2Ｒ (0～31)
3	シンクロ (0,1)	ＲＲ (0～15)
4	スピード (0～255)	D1Ｌ (0～15)
5	ＰＭＤ (0～127)	ＴＬ (0～127)
6	ＡＭＤ (0～7)	ＫＳ (0～3)
7	ＰＭＳ (0～3)	ＭＵＬ (0～15)
8	ＡＭＳ (0～3)	ＤT1 (0～7)
9	Ｌ，ＲＰＡＮ (0～3)	ＤT2 (0～3)
10		ＡＭＳイネーブル (0,1)

なお、オペレータ1はM1（モジュレータ1）、オペレータ2はC1（キャリア1）、オペレータ3はM2（モジュレータ2）、およびオペレータ4はC2（キャリア2）に対応しています。

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。

参照関数

m_vget

m_ycom

レベル3

書　式

```
#include <music3.h>

int m_ycom(md);
int md;  /* Ｙコマンドの使用モード */
```

機　能

md=0でＹコマンドをＭＩＤＩのコントロールデータと解釈します。
md=1でＹコマンドをＯＰＭレジスタの書き換えと解釈します。
mdに-1または'NASI'を指定した場合は、現在のモードを返します。
(デフォルト=1)

戻り値

正常終了の場合は0を返します。
エラーの場合は-1を返し、そのエラーコードはm_errget関数で調べることができます。
mdに-1または'NASI'を指定した場合は、現在のモードを返します。
(デフォルト=1)

1.10 IOCSコール一覧

● ＩＯＣＳコール（アルファベット順）

番号	名　前	機　　能	頁
$01	M_ALLOC	指定トラックのトラックバッファの確保	94
$18	M_ANTOFF	指定チャンネルへオールノートオフ出力	107
$02	M_ASSIGN	トラックを指定	95
$0C	M_ATOI	指定トラックバッファの先頭アドレス取得	100
$45	M_BEND	指定チャンネルのピッチベンド設定	127
$10	M_CHAN	出力チャンネルの一時変更	101
$0B	M_CONT	指定チャンネルの演奏再開	100
$3D	M_CTRLRES	指定チャンネルへ各種のオフ出力	122
$19	M_DIROUT	ＭＩＤＩデータの出力	107
$1A	M_ENABLE	指定チャンネルの発音許可	108
$35	M_END	停止小節の設定	117
$07	M_FREE	指定トラックバッファの残りバイト数の取得	98
$2F	M_FREEA	ﾄﾞﾗｲﾊﾞ起動時のﾄﾗｯｸﾊﾞｯﾌｧ全バイト数を取得	114
$2E	M_IFCHK	ＭＩＤＩインターフェイスの有無を検出	114
$00	M_INIT	トラックバッファの初期化	93
$3F	M_MDREG	ＭＩＤＩ制御ＩＣのレジスタに値をセット	124
$32	M_MEAS	演奏中の小節番号取得	115
$21	M_METER	曲の拍子設定	111
$46	M_MOD	指定チャンネルのモジュレーション設定	127
$36	M_MODSNS	ＦＭ音源のモジュレーションの深さ設定	117
$44	M_MSTVOL	マスターボリュームの設定	126
$1C	M_NTOFF	指定チャンネルへノートオフの出力	109
$1B	M_NTON	指定チャンネルへノートオンの出力	108
$47	M_OPMEXC	ＦＭ音源のパラメータセットの選択	128
$48	M_OPMLFQ	ＦＭ音源のＬＦＯ発振周波数の設定	129
$3E	M_OPMREG	指定のＯＰＭレジスタに値をセット	123
$30	M_OUT	１バイトデータをＭＩＤＩへ出力	115
$13	M_PAN	指定チャンネルのパンポット設定	104
$4B	M_PANFLT	コマンドのモードの指定	132
$3B	M_PCMBSY	ＰＣＭの実行状態取得	120
$39	M_PCMGET	登録されているＰＣＭデータの取得	119
$3A	M_PCMLEN	指定のＰＣＭデータの取得	119
$3C	M_PCMON	指定のＰＣＭデータを再生	121
$38	M_PCMREC	外部入力音の録音	118
$2A	M_PCMSET	指定の音程にＰＣＭ音を登録	112
$49	M_PGMFLT	ＭＭＬ音色切り替えコマンドの出力選択	130
$08	M_PLAY	指定チャンネルの演奏開始	98
$43	M_PNMGET	ＰＣＭ音色名の取得	126
$42	M_PNMSET	ＰＣＭ音色名の登録	125
$11	M_PROG	指定チャンネルの音色番号設定	102
$29	M_SNDSET	ＳＮＤファイルのデータをＦＭ音源用に登録	111
$34	M_START	演奏開始小節の設定	116
$09	M_STAT	指定チャンネルの現在の演奏状態を返す	99
$0A	M_STOP	指定チャンネルの演奏停止	99
$20	M_SYNC	同期モード設定	110

番号	名　前	機　　能	頁
$2B	M_SYSCH	チャンネル番号の割り当て変更	113
$05	M_TEMPO	曲のテンポを指定	97
$41	M_TNMGET	ＦＭ音源の音色名の取得	125
$40	M_TNMSET	ＦＭ音源の音色名の登録	124
$06	M_TRK	指定トラックにＭＭＬデータをセット	97
$15	M_TRNS	指定チャンネルの音程シフト	106
$0F	M_USE	指定トラックバッファの使用容量の取得	101
$14	M_VEL	指定チャンネルのベロシティ設定	105
$5E	M_VERSION	OPMDRV3.Xのバージョンと作成日付の取得	133
$03	M_VGET	ＦＭ音源の音色データの取得	95
$12	M_VOL	指定チャンネルのボリューム設定	103
$4A	M_VOLFLT	ＭＭＬボリューム設定コマンドの出力選択	131
$04	M_VSET	ＦＭ音源の音色を設定	96
$33	M_YCOM	Ｙコマンドの解釈	116

1.11 IOCSコールリファレンス

「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ　ｖｅｒ２．１」では、ＯＰＭＤＲＶ３．Ｘにより ＦＭ音源／ＭＩＤＩ／ＡＤＰＣＭを同時に制御できます。
また、次のようにＩＯＣＳコールを利用してアセンブラからその機能を利用することも可能です。

```
moveq.l  #OPMコール番号, D1
moveq.l  #$F0, D0
trap     #15
```

（ＯＰＭＤＲＶ３．Ｘ用のＩＯＣＳコール番号は全て＄Ｆ０です。）

例）　トラックデータの初期化

```
move.w    #$01,D2
moveq.l   #$00,D1
moveq.l   #$F0,D0
trap      #15
```

または、マクロを使用して

```
include   iocscall.mac
    :
    :
move.w    #1,D2
moveq.l   #_M_INIT
IOCS      _OPMDRV
```

ＩＯＣＳコールの詳細説明は、「プログラマーズマニュアル 第３章 ＩＯＣＳコール」を参照してください。

M_INIT

●ＯＰＭコール番号　：　$00　　ＯＰＭコマンド　：　(In)

機　能	トラックバッファの初期化
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$00　　ＯＰＭコール番号 D2. W　　初期化モード
リターン	D0. L　　エラーコード D0. L =0 ……… 正常終了
解　説	現在のトラックデータを全てクリアし、各トラックバッファを４バイトに設定します。 また、各チャンネルのパラメータは、以下のように初期化されます。

トラック番号	＝	チャンネル番号
音色番号	＝	１
ボリューム	＝	６４
パンポット	＝	６４
ベロシティ	＝	１２７
トランスポーズ	＝	２４
拍子	＝	４／４
テンポ	＝	１２０
同期モード	＝	内部同期

入力のD2. Wは、音色データの初期化モードです。
D2. Wに０を指定し場合は、音色データを初期化せず、１の場合は初期化します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_ALLOC

●ＯＰＭコール番号　：　$01　　ＯＰＭコマンド　：　(M t,s)

機　能

指定トラックのトラックバッファの設定

入　力

D0. L	$ＦＯ	ＩＯＣＳコール番号
D1. L	$01	ＯＰＭコール番号
D2. L	トラック番号×65536＋バッファサイズ	

リターン

D0. L　　エラーコード

D0. L＝0 ……… 正常終了

解　説

指定のトラックのバッファサイズを設定します。
全トラックバッファ内のデータは、クリアされます。
なお、総トラックのバッファサイズ（それぞれのトラックに割り当てられているトラックバッファの総合計）は、ＣＯＮＦＩＧ．ＳＹＳファイル内のＯＰＭＤＲＶ３．Ｘの後に指定して、再度システム起動することにより変更できます。

```
DEVICE=¥SYS¥OPMDRV3.X #nnn
```

(nnn：総トラックバッファサイズ［単位：ＫＢ］)

入力のD2. Lは、上位１６ビットがトラック番号（1～80）、下位１６ビットがバッファサイズ（1～65535）です。
バッファサイズは、バイト数で指定し、１～３を指定した場合でも４バイト確保します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_ASSIGN

●ＯＰＭコール番号　：　$02　　ＯＰＭコマンド　：　(A c,t)

機　能	指定チャンネルのトラック指定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$02　　ＯＰＭコール番号 D2. L　　チャンネル番号×65536＋トラック番号
リターン	D0. L　　エラーコード D0. L＝0………正常終了
解　説	指定のチャンネルで使用するトラックを設定します。 複数のチャンネルに同一のトラックを割りあてると、それぞれ同じ楽譜データを演奏します。 また、複数のトラックを１つのチャンネルに割りあてると、最も新しく割りふられたトラックの楽譜データを演奏します。 デフォルト値は、チャンネル１～２５が、それぞれトラック１～２５に対応します。 入力のD2. Lは、上位１６ビットがチャンネル番号（1～25）、下位１６ビットがトラック番号（1～80）です。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_VGET

●ＯＰＭコール番号　：　$03　　ＯＰＭコマンド　：　なし

機　能	ＦＭ音源の音色データ取得
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$03　　ＯＰＭコール番号 D2. L　　音色番号(1～200) A1. L　　音色データを格納するためのバッファ先頭アドレス
リターン	D0. L　　エラーコード D0. L＝0………正常終了 (A1)　　音色データ
解　説	ＦＭ音源の指定した音色番号のデータをバッファにセットします。 入力のD2. Lは、音色番号で１～６８は登録済です。 入力のA1. Lは、音色データを格納するためのバッファ先頭アドレスでバッファサイズは５５バイト必要です。 なお、音色データの内部形式は、M_VSETを参照してください。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_VSET

●ＯＰＭコール番号　：　$04　　ＯＰＭコマンド　：　(V n1,n2,data..)

機　能	ＦＭ音源の音色を設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$04　　ＯＰＭコール番号 D2. L　　音色番号(1～200) A1. L　　音色データの先頭アドレス（55バイト）
リターン	D0. L　　エラーコード D0. L＝0………正常終了

解　説

ＦＭ音源の指定した音色番号に音色データを設定します。
入力のD2. Lは、音色番号で１～６８は登録済です。
入力のA1. Lは、音色データの先頭アドレスです。
サイズは５５バイトで、設定内容は次の通りです。
音色データの内部形式

	0	1～4(ｵﾍﾟﾚｰﾀ)
0	フィードバック／アルゴリズム(0～63)	ＡＲ　(0～31)
1	スロットマスク　(0～15)	D1Ｒ　(0～31)
2	ウェーブフォーム　(0～3)	D2Ｒ　(0～31)
3	シンクロ　(0,1)	ＲＲ　(0～15)
4	スピード　(0～255)	D1Ｌ　(0～15)
5	ＰＭＤ　(0～127)	ＴＬ　(0～127)
6	ＡＭＤ　(0～7)	ＫＳ　(0～3)
7	ＰＭＳ　(0～3)	ＭＵＬ　(0～15)
8	ＡＭＳ　(0～3)	ＤＴ1　(0～7)
9	Ｌ，ＲＰＡＮ　(0～3)	ＤＴ2　(0～3)
10		ＡＭＳイネーブル（0,1）

なお、オペレータ1はM1（モジュレータ1）、オペレータ2はC1（キャリア1）、オペレータ3はM2（モジュレータ2）、およびオペレータ4はC2（キャリア2）に対応しています。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_TEMPO

●ＯＰＭコール番号　：　$05　　ＯＰＭコマンド　：　(0 n)

機　能

曲のテンポを指定

入　力

D0. L	＄F0	ＩＯＣＳコール番号
D1. L	$05	ＯＰＭコール番号
D2. L	テンポデータ(20～300)	

リターン

D2. L＝－１：
　D0. L　　現在のテンポ
D2. L≠－１：
　D0. L　　エラーコード
　　D0. L＝0 ……… 正常終了

解　説

ＦＭ音源およびＭＩＤＩの演奏時のテンポを指定します。
入力のD2. Lは、テンポで１分間に４分音符をいくつ演奏できるかで指定します。
デフォルトは１２０です。
－１を指定した場合は、現在のテンポを返します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_TRK

●ＯＰＭコール番号　：　$06　　ＯＰＭコマンド　：　(T n)

機　能

指定トラックにＭＭＬデータをセット

入　力

D0. L	＄F0	ＩＯＣＳコール番号
D1. L	$06	ＯＰＭコール番号
D2. L	トラック番号(1～80)	
A1. L	ＭＭＬデータ先頭アドレス	

リターン

D0. L　　エラーコード
　D0. L＝0 ……… 正常終了

解　説

指定のトラックへＭＭＬデータをセットします。
入力のA1. ＬはＭＭＬデータの先頭アドレスで、終端はヌル文字 (0) です。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_FREE

●ＯＰＭコール番号　：　$07　　ＯＰＭコマンド　：　なし

機　能	指定トラックバッファの残りバイト数の取得
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$07　　ＯＰＭコール番号 D2. L　　トラックNo.(1～80)
リターン	D0. L　　残りバイト数
解　説	指定のトラックバッファの残りバイト数を返します。

M_PLAY

●ＯＰＭコール番号　：　$08　　ＯＰＭコマンド　：　(P n1,n2,...)

機　能	指定チャンネルの演奏開始
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$08　　ＯＰＭコール番号 D2. L　　チャンネルのビットパターン
リターン	D0. L　　エラーコード D0. L＝0……正常終了
解　説	指定のチャンネルの演奏を開始します。 入力のD2. Lは、演奏の開始を指定するチャンネルのビットパターンで、ビット０～２４がチャンネル１～２５に対応しており、演奏を開始したいチャンネルに対応したビットに１を指定します（他のビットは常に０）。 －１を指定した場合は、全てのチャンネルの演奏を開始します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_STAT

●ＯＰＭコール番号　：　$09　　ＯＰＭコマンド　：　なし

機　能　指定チャンネルの現在の演奏状態を返す

入　力

D0. L　　$F0　　ＩＯＣＳコール番号
D1. L　　$09　　ＯＰＭコール番号
D2. L　　チャンネル番号(1～25)

リターン

D2. L =1～25　:
　D0. L =0 ………… 停止中
　D0. L =1 ………… 演奏中
D2. L =-1　:
　D0. L　演奏状態を示すチャンネルのビットパターン

解　説

指定のチャンネルの現在の演奏状態を返します。
入力のD2. Lに－１を指定した場合は、全てのチャンネルの演奏状態をビットパターンで返します。
ビット０～２４がチャンネル１～２５に対応しており、チャンネルに対応したビットに演奏状態をセットします（他のビットは常に０）。

M_STOP

●ＯＰＭコール番号　：　$0A　　ＯＰＭコマンド　：　(S n1,n2,...)

機　能　指定チャンネルの演奏停止

入　力

D0. L　　$F0　　ＩＯＣＳコール番号
D1. L　　$0A　　ＯＰＭコール番号
D2. L　　チャンネルのビットパターン

リターン

D0. L　　エラーコード
　D0. L =0 ……… 正常終了

解　説

指定されたチャンネルの演奏を停止します。
入力のD2. Lは、演奏の停止を指定するチャンネルのビットパターンで、ビット０～２４がチャンネル１～２５に対応しており、演奏を停止させたいチャンネルに対応したビットに１を指定します（他のビットは常に０）。
－１を指定した場合は、全てのチャンネルを停止。
リターンのD0. Lのエラーコードについては、「１．１３　エラーコード一覧」を参照してください。

M_CONT

●ＯＰＭコール番号　：　$0B　ＯＰＭコマンド　：　(C)

機　能

指定チャンネルの演奏再開

入　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$0B	ＯＰＭコール番号
D2. L	チャンネルのビットパターン	

リターン

D0. L　　エラーコード

D0. L＝0………正常終了

解　説

指定されたチャンネルの演奏を再開します。
入力のD2. Lは、演奏の再開を指定するチャンネルのビットパターンで、ビット０～２４がチャンネル１～２５に対応しており、演奏を再開させたいチャンネルに対応したビットに１を指定します（他のビットは常に０）。
－１を指定した場合は、全てのチャンネルを再開。
リターンのD0. Lのエラーコードについては、「１．１３　エラーコード一覧」を参照してください。

M_ATOI

●ＯＰＭコール番号　：　$0C　ＯＰＭコマンド　：　なし

機　能

指定トラックバッファの先頭アドレス取得

入　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$0C	ＯＰＭコール番号
D2. L	トラック番号（1～80）	

リターン

D0. L　　トラックバッファ先頭アドレス

解　説

指定のトラックのトラックバッファ先頭アドレスを返します。
トラックバッファは、ＯＰＭＤＲＶ３．Ｘの内部バッファのため、スーパーバイザー領域です。
データのアクセスには、注意してください。

M_USE

●ＯＰＭコール番号　：　$0F　　ＯＰＭコマンド　：　なし

機　　能	指定トラックバッファの使用容量の取得
入　　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$0F　　ＯＰＭコール番号 D2. L　　トラック番号(1～80)
リターン	D0. L　　バッファ使用容量
解　　説	指定のトラックバッファにおける現在の使用容量を返します。 入力のD2. Lに０を指定した場合は、８０トラック全ての合計使用容量を返します。

M_CHAN

●ＯＰＭコール番号　：　$10　　ＯＰＭコマンド　：　(N c,c)

機　　能	出力チャンネルの一時変更
入　　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$10　　ＯＰＭコール番号 D2. L　チャンネル番号×65536＋出力チャンネル番号
リターン	D2. W＝-1　： D0. L　現在の出力チャンネル番号 D2. W≠-1　： D0. L　　エラーコード D0. L＝0 ……… 正常終了
解　　説	指定のチャンネルの出力チャンネルを一時的に変更します。 ＭＭＬデータの@Nとは異なり、変更前のチャンネルの各パラメータ（トラック番号、音色番号、ボリュームなど）を引き継ぎます。 入力のD2. Lは、上位１６ビットがチャンネル番号（1～25）、下位１６ビットが出力チャンネル番号（1～25）です。 出力チャンネルに－１を指定した場合は、現在の設定値を返します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PROG

●ＯＰＭコール番号　：　$11　　ＯＰＭコマンド　：　(G c,n)

機　　能　指定チャンネルの音色番号設定

入　　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$11	ＯＰＭコール番号
D2. L	チャンネル番号×65536＋音色番号	

リターン

D2. W＝-1　：
　D0. L　現在の音色番号
D2. W≠-1　：
　D0. L　エラーコード
　　D0. L＝0 ……… 正常終了

解　　説

指定のチャンネルの音色番号を設定します。
入力のD2. Lは、上位１６ビットがチャンネル番号（1～25）、下位１６ビットが音色番号（ＦＭ音源は1～200、ＭＩＤＩは1～128）です。
チャンネル番号に－１を指定した場合は、全てのチャンネルに同じ音色番号を設定します。
音色番号に－１を指定した場合は、現在の設定値を返します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_VOL

●ＯＰＭコール番号　：　$12　　ＯＰＭコマンド　：　(L c,n)

機　　能　指定チャンネルのボリューム設定

入　　力

D0.L　　$F0　　ＩＯＣＳコール番号
D1.L　　$12　　ＯＰＭコール番号
D2.L　　チャンネル番号×65536＋ボリュームデータ

リターン

D2.W＝-1　：
　D0.L　現在のボリューム
D2.W≠-1　：
　D0.L　エラーコード
　D0.L＝0 ……… 正常終了

解　　説

指定のチャンネルにボリュームデータを設定します。
入力のD2.Lは、上位１６ビットがチャンネル番号（1～25）、下位１６ビットがボリュームデータ（0～127）です。
チャンネル番号に－１を指定した場合は、全てのチャンネルに同じボリュームデータを設定します。
ボリュームデータに－１を指定した場合は、現在の設定値を返します。
リターンのD0.Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PAN

●ＯＰＭコール番号　：　$13　　ＯＰＭコマンド　：　(B c,n)

機　能

指定チャンネルのパンポット設定

入　力

D0.L　　$F0　　ＩＯＣＳコール番号
D1.L　　$13　　ＯＰＭコール番号
D2.L　　チャンネル番号×65536＋パンポットデータ

リターン

D2.W＝-1　：
　D0.L　現在のパンポットデータ
D2.W≠-1　：
　D0.L　　エラーコード
　　D0.L＝0 ……… 正常終了

解　説

指定のチャンネルのパンポットデータを設定します。
入力のD2.Lは、上位１６ビットがチャンネル番号（1～25）、下位１６ビットがパンポットデータ（0～127）です。
チャンネル番号に－１を指定した場合は、全てのチャンネルに同じパンポットデータを設定します。
パンポットデータのデフォルト値は６４です。
パンポットデータに－１を指定した場合は、現在の設定値を返します。

パンポットデータ
　　０：　ライト（右端）
　　⋮
　６４：　センター（中央）
　　⋮
１２７：　レフト（左端）

リターンのD0.Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_VEL

●ＯＰＭコール番号　：　$14　　ＯＰＭコマンド　：　(U c,n)

機　能	指定チャンネルのベロシティを設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$14　　ＯＰＭコール番号 D2. L　　チャンネル番号×65536＋ベロシティデータ
リターン	D2. W＝-1　： 　D0. L　現在のベロシティデータ D2. W≠-1　： 　D0. L　エラーコード 　　D0. L＝0 ……… 正常終了
解　説	指定のチャンネルのベロシティデータを設定します。 入力のD2. Lは、上位１６ビットがチャンネル番号（1〜25）、下位１６ビットがベロシティデータ（0〜127）です。 チャンネル番号に－１を指定した場合は、全てのチャンネルに同じベロシティデータを設定します。 ベロシティデータのデフォルト値は１２７です。 ベロシティデータに－１を指定した場合は、現在の設定値を返します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_TRNS

●ＯＰＭコール番号　：　$15　　ＯＰＭコマンド　：　(R c,n)

機　能	指定チャンネルの音程シフト
入　力	D0. L　$F0　ＩＯＣＳコール番号 D1. L　$15　ＯＰＭコール番号 D2. L　チャンネル番号×65536＋トランスポーズデータ
リターン	D2. W＝-1　： D0. L　現在のトランスポーズデータ D2. W≠-1　： D0. L　エラーコード D0. L＝0 ……… 正常終了
解　説	指定のチャンネルの音程を±2オクターブの範囲でシフトします。 入力のD2. Lは、上位１６ビットがチャンネル番号（1～25）、下位１６ビットがトランスポーズデータ（0～48）です。 チャンネル番号に－１を指定した場合は、全てのチャンネルに同じトランスポーズデータを設定します。 トランスポーズデータのデフォルト値は６４です。 トランスポーズデータに－１を指定した場合は、現在の設定値を返します。 リターンのD0. Lのエラーコードについては、「1．１３ エラーコード一覧」を参照してください。

M_ANTOFF

●ＯＰＭコール番号　：　$18　　ＯＰＭコマンド　：　(Ｊｃ)

機能	指定チャンネルへオールノートオフ出力		
入力	D0. L	$F0	ＩＯＣＳコール番号
	D1. L	$18	ＯＰＭコール番号
	D2. L	チャンネル番号（1～25）	
リターン	D0. L	エラーコード	
	D0. L＝0………正常終了		

解説

指定のチャンネルへオールノートオフを出力します。
入力のD2. Lに－１を指定した場合は、全てのチャンネルへノートオフを出力します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_DIROUT

●ＯＰＭコール番号　：　$19　　ＯＰＭコマンド　：　(X n1,n2...n512)

機能	ＭＩＤＩデータの出力		
入力	D0. L	$F0	ＩＯＣＳコール番号
	D1. L	$19	ＯＰＭコール番号
	D2. L	データサイズ	
	A1. L	ＭＩＤＩデータの先頭アドレス	
リターン	D0. L	エラーコード	
	D0. L＝0………正常終了		

解説

ＭＩＤＩデータをデータサイズ分ＭＩＤＩへ出力します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_ENABLE

●ＯＰＭコール番号　：　$1A　ＯＰＭコマンド　：　(E n1,n2...)

機　能　指定チャンネルの発音許可

入　力

D0.L　$F0　ＩＯＣＳコール番号
D1.L　$1A　ＯＰＭコール番号
D2.L　チャンネルのビットパターン

リターン

D2.L＝-1　：
　D0.L　現在の許可状態を示すチャンネルのビットパターン
D2.L≠-1　：
　D0.L　エラーコード
　　D0.L＝0………正常終了

解　説

チャンネルのビットパターンで示されるチャンネルの演奏を、許可（ビット＝1）または禁止（ビット＝0）します。
M_STOPとは異なり、内部的にはノート以外のデータは、継続しれています（ミュート状態）。
入力のD2.Lは、発音許可を指定するチャンネルのビットパターンで、ビット０～２４がチャンネル１～２５に対応しており、発音を許可したいチャンネルに対応したビットに1を指定します（他のビットは常に０）。
－１を指定した場合は、現在の許可状態を示すチャンネルのビットパターンを返します。
リターンのD0.Lのエラーコードについては、「1．１３　エラーコード一覧」を参照してください。

M_NTON

●ＯＰＭコール番号　：　$1B　ＯＰＭコマンド　：　なし

機　能　指定チャンネルへノートオン出力

入　力

D0.L　$F0　ＩＯＣＳコール番号
D1.L　$1B　ＯＰＭコール番号
D2.L　ﾁｬﾝﾈﾙ番号×65536＋ノート番号×256＋ﾍﾞﾛｼﾃｨﾃﾞｰﾀ

リターン

D0.L　エラーコード
　D0.L＝0………正常終了

解　説

指定のチャンネルへノートオンを出力します。
入力のD2.Lは、上位16ビットがチャンネル番号(1～25)、ビット8～15がノート番号(0～127)、下位8ビットがベロシティ(0～127)です。
リターンのD0.Lのエラーコードについては、「1．１３　エラーコード一覧」を参照してください。

M_NTOFF

●ＯＰＭコール番号　：　$1C　ＯＰＭコマンド　：　なし

機　能	指定チャンネルへノートオフ出力
入　力	D0.L　$F0　ＩＯＣＳコール番号 D1.L　$1C　ＯＰＭコール番号 D2.L　ﾁｬﾝﾈﾙ番号×65536＋ノート番号×256＋ﾍﾞﾛｼﾃｨﾃﾞｰﾀ
リターン	D0.L　エラーコード D0.L＝0 ……… 正常終了
解　説	指定のチャンネルへノートオフを出力します。 入力のD2.Lは、上位１６ビットがチャンネル番号（1～25）、ビット8～15がノート番号（0～127）、下位８ビットがベロシティ（0～127）です。 リターンのD0.Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_SYNC

●ＯＰＭコール番号　：　$20　　ＯＰＭコマンド　：　(Z n)

機　能　同期モード設定

入　力

D0. L　　$F0　　ＩＯＣＳコール番号
D1. L　　$20　　ＯＰＭコール番号
D2. W　　同期モード

リターン

D2. W＝-1　　:
　D0. L　現在の同期モード
D2. W≠-1　　:
　D0. L　　エラーコード
　　D0. L＝0 ……… 正常終了

解　説

ＭＩＤＩクロック同期モードを設定します。
入力のD2. Wは同期モードで、設定内容は次の通りです。

D2. W＝0　………… 内部のＭＩＤＩクロックに合わせて演奏
　　＝1　………… 外部のＭＩＤＩクロックに合わせて演奏
　　＝2　………… 外部のＦＳＫクロックに合わせて演奏
　　＝－1 ………… 現在の状態をD0. Lに返します。

外部同期にする場合は、同期モードに1を指定し、M_SYNC実行後M_PLAYで待機状態にセットしてＭＩＤＩのスタートコマンド（$FA）を待ちます。
リターンのD0. Lのエラーコードについては、「1．13 エラーコード一覧」を参照してください。

M_METER

●ＯＰＭコール番号　：　$21　　ＯＰＭコマンド　：　(Q n1,n2)

機　能	曲の拍子設定		
入　力	D0. L	$F0	ＩＯＣＳコール番号
	D1. L	$21	ＯＰＭコール番号
	D2. L	拍子の分子×65536＋拍子の分母	
リターン	D0. L	エラーコード	
	D0. L＝0 ……… 正常終了		

解　説

曲の拍子を設定します。
入力のD2. Lは、上位１６ビットが拍子の分子（1～16）、下位１６ビットが拍子の分母（2,4,8,16）です。
デフォルト値は４／４拍子に設定されています。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_SNDSET

●ＯＰＭコール番号　：　$29　　ＯＰＭコマンド　：　無し

機　能	ＳＮＤファイル形式のデータをＦＭ音源用に登録		
入　力	D0. L	$F0	ＩＯＣＳコール番号
	D1. L	$29	ＯＰＭコール番号
	D2. L	データサイズ（最大16000バイト）	
	A1. L	データの先頭アドレス	
リターン	なし（D0. Lの内容は保証されません）		

解　説

SOUND PRO-68K のＳＮＤファイル形式のデータをＦＭ音源用の音色データとして登録します。
入力のD2. Lは、データサイズで１音色８０バイトを最大２００音色分（＝16000バイト）まで指定できます。
入力のA1. Lは、ＳＮＤファイル形式のデータが格納されているバッファの先頭アドレスです。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PCMSET

●ＯＰＭコール番号　：　$2A　ＯＰＭコマンド　：　無し

機　能　指定の音程にＰＣＭ音を登録

入　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$2A	ＯＰＭコール番号
D2. L	データサイズ（最大32767バイト）	
D3. L	ノート番号×65536＋サンプリング周波数	
A1. L	ＰＣＭデータの先頭アドレス	

リターン　D0. L　エラーコード

D0. L＝0………正常終了

解　説

指定の音程に音を登録します。使用できる音程は'C-2'から'G#8'（ノート番号0～127に対応）の１２８音です。
入力のD2. Lは、ＰＣＭデータのバイト数で－１を指定した場合は、ＰＣＭバッファをクリアします。
入力のD3. Lは、上位１６ビットがノート番号（0～127）、下位１６ビットがサンプリング周波数（0～4）です。

サンプリング周波数

0………３．９ＫＨｚ
1………５．２ＫＨｚ
2………７．８ＫＨｚ
3………10．４ＫＨｚ
4………15．６ＫＨｚ

リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_SYSCH

●ＯＰＭコール番号　：　$2Ｂ　ＯＰＭコマンド　：　(/OMP)

機　能　チャンネル番号の割り当て変更

入　力
D0. L　$F0　ＩＯＣＳコール番号
D1. L　$2Ｂ　ＯＰＭコール番号
D2. W　割り当てモード（0〜5）

リターン
D2. W＝-1　：
　D0. L　割り当てモード
D2. W≠-1　：
　D0. L　エラーコード
　　D0. L＝0 ……… 正常終了

解　説
各音源に対するチャンネル番号の割り当てモードを設定します。
入力のD2. Wは割り当てモードで、設定内容は次の通りです。
割り当てモード

```
0="OPM"・・・・・・・OPM=ch1〜8     PCM=ch9        MIDI=ch10〜25
1="OMP"・・・・・・・OPM=ch1〜8     MIDI=ch9〜24   PCM=ch25
2="MOP"・・・・・・・MIDI=ch1〜16   OPM=ch17〜24   PCM=ch25
3="MPO"・・・・・・・MIDI=ch1〜16   PCM=ch17       OPM=ch18〜25
4="POM"・・・・・・・PCM=ch1        OPM=ch2〜9     MIDI=ch10〜25
5="PMO"・・・・・・・PCM=ch1        MIDI=ch2〜17   OPM=ch18〜25
```

入力のD2. Wが－１の場合は現在のモードをD0. に返します。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_IFCHK

●ＯＰＭコール番号　：　$2C　　ＯＰＭコマンド　：　無し

機　能	ＭＩＤＩインターフェイスの有無を検出
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$2C　　ＯＰＭコール番号
リターン	D0. L　　エラーコード D0. L＝0………ある D0. L＝-1………ない
解　説	ＭＩＤＩインターフェイスの有無をチェックします。

M_FREEA

●ＯＰＭコール番号　：　$2F　　ＯＰＭコマンド　：　無し

機　能	ドライバ起動時のトラックバッファ全バイト数を取得
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$2F　　ＯＰＭコール番号
リターン	D0. L　　全トラックバッファのバイト数
解　説	ドライバ起動時に確保したトラックバッファの全バイト数を返します。

M_OUT

●ＯＰＭコール番号　：　$30　　ＯＰＭコマンド　：　無し

機　能	１バイトのデータをＭＩＤＩへ出力
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$30　　ＯＰＭコール番号 D2. B　　出力データ（0～225）
リターン	なし（D0. Lの内容は保証されません）
解　説	１バイトデータをＭＩＤＩに出力します。

M_MEAS

●ＯＰＭコール番号　：　$32　　ＯＰＭコマンド　：　無し

機　能	演奏中の小節番号取得
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$32　　ＯＰＭコール番号
リターン	D0. L　　小節番号(0～32767)
解　説	現在演奏中の小節番号を返します。 ＭＭＬデータ"|:"," D.S." などの繰り返し記号の場合はそのまま加算されますので、実際の小節番号とは異なる場合があります。

M_YCOM

●ＯＰＭコール番号　：　$33　　ＯＰＭコマンド　：　(/Y n)

機　　能　　Ｙコマンドの解釈

入　　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$33	ＯＰＭコール番号
D2. L	使用モード	

リターン

D2. L =-1　：
　D0. L　現在の使用モード
D2. L ≠-1　：
　D0. L　不定

解　　説

Ｙコマンドの使用モードを設定します。
M_TRK実行前に設定してください。
入力のD2. Lは、Ｙコマンドの使用モードです。
D2. Lに０を指定した場合は、ＹコマンドをＭＩＤＩのコントロールデータとして解釈し、１の場合はＯＰＭレジスタ書き換えと解釈します。

M_START

●ＯＰＭコール番号　：　$34　　ＯＰＭコマンド　：　(PM n)

機　　能　　演奏開始小節の設定

入　　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$34	ＯＰＭコール番号
D2. W	小節番号 (0～32767)	

リターン

D0. L　　エラーコード
　D0. L =0 ……… 正常終了

解　　説

Ｍ＿ＰＬＡＹで演奏を開始する小節位置を設定します。
全てのチャンネルのＭＭＬデータ"|:"," D.S.",":|"," D.C."などの数や位置が同じでないと正常な演奏ができません。
なお、一度設定された小節番号は、演奏が終わるとクリアされます。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_END

●ＯＰＭコール番号　:　$35　　ＯＰＭコマンド　:　(SM n)

機　能	演奏停止小節の設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$35　　ＯＰＭコール番号 D2. W　　小節番号（1～32767）
リターン	D0. L　　エラーコード D0. L＝0……正常終了
解　説	演奏を停止する小節位置を設定します。 全てのチャンネルのＭＭＬデータ"|:"," D.S."," :|"," D.C."などの数や位置が同じでないと正常な演奏ができません。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_MODSNS

●ＯＰＭコール番号　:　$36　　ＯＰＭコマンド　:　無し

機　能	ＦＭ音源のモジュレーションの深さ設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$36　　ＯＰＭコール番号 D2. L　　モジュレーションの深さ（0～127）
リターン	D2. W＝-1　: D0. L　現在のモジュレーションの深さ D2. W≠-1　: D0. L　　エラーコード D0. L＝0……正常終了
解　説	ＦＭ音源全体にかかるモジュレーションの深さを設定します。 入力のD2. Lは、ＦＭ音源のモジュレーションの深さで、－１を指定した場合は、現在の設定値を返します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PCMREC

●ＯＰＭコール番号　：　$38　　ＯＰＭコマンド　：　無し

機　　能

外部入力音のＰＣＭ録音

入　　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$38	ＯＰＭコール番号
D2. W	サンプリング周波数（0～4）	
D3. L	バッファサイズ（最大32767バイト）	
A1. L	ＰＣＭデータバッファの先頭アドレス	

リターン

D0. L　　エラーコード

D0. L＝0……正常終了

D0. L＝-1……タイムオーバー

（A1）　　ＰＣＭデータ

解　　説

オーディオ入力端子より入力された音をＰＣＭ録音し、バッファにセットします。

４秒以内に音が入力されなければ、D0. Lに－１を返します。

録音の頭出しは自動処理されます。（音が入力された時点で録音を開始します。）

入力のD2. Wはサンプリング周波数で設定内容は、次の通りです。

サンプリング周波数

0……３．９ＫＨｚ（１秒につき1950バイト）

1……５．２ＫＨｚ（１秒につき2600バイト）

2……７．８ＫＨｚ（１秒につき3900バイト）

3……10．４ＫＨｚ（１秒につき5200バイト）

4……15．６ＫＨｚ（１秒につき7800バイト）

リターンのD0. Lのエラーコードについては、「1．13　エラーコード一覧」を参照してください。

M_PCMGET

●ＯＰＭコール番号　：　$39　　ＯＰＭコマンド　：　無し

機　能	登録されているＰＣＭデータの取得		
入　力	D0. L	$F0	ＩＯＣＳコール番号
	D1. L	$39	ＯＰＭコール番号
	D2. W	ノート番号（0～127）	
	D3. L	バッファサイズ（バイト数）	
	A1. L	ＰＣＭデータバッファの先頭アドレス	
リターン	D0. L	読み込まれたデータサイズ	
	(A1)	ＰＣＭデータ	

解　説

現在ドライバ内に登録されている指定ノート番号に対応するＰＣＭデータをバッファにセットします。

M_PCMLEN

●ＯＰＭコール番号　：　$3A　　ＯＰＭコマンド　：　無し

機　能	指定のＰＣＭデータのサイズを取得		
入　力	D0. L	$F0	ＩＯＣＳコール番号
	D1. L	$3A	ＯＰＭコール番号
	D2. W	ノート番号（0～127）	
リターン	D0. L	ＰＣＭデータのサイズ（バイト数）	

解　説

現在ドライバ内に登録されている指定ノート番号に対応するＰＣＭデータのサイズを返します。

M_PCMBSY

●ＯＰＭコール番号　：　$3Ｂ　ＯＰＭコマンド　：　無し

機　能	ＰＣＭの実行状態を返す
入　力	D0.L　$F0　ＩＯＣＳコール番号 D1.L　$3Ｂ　ＯＰＭコール番号
リターン	D0.L　実行状態 D0.L＝$00………何も実行していない D0.L＝$02………出力中（ADPCMOUT,M_PCMONを実行中） D0.L＝$04………入力中（ADPCMINP,M_PCMRECを実行中） D0.L＝$12………出力中（ADPCMAOTを実行中） D0.L＝$14………入力中（ADPCMAINを実行中） D0.L＝$22………出力中（ADPCMLOTを実行中） D0.L＝$24………入力中（ADPCMLINを実行中）
解　説	現在のＰＣＭの実行状態を返します。

M_PCMON

●ＯＰＭコール番号　：　$3C　ＯＰＭコマンド　：　無し

機　能	指定のＰＣＭデータを再生
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$3C　　ＯＰＭコール番号 D2. L　　ﾉｰﾄ番号×65536+ｻﾝﾌﾟﾘﾝｸﾞ周波数×256+音声出力ﾓｰﾄﾞ
リターン	なし
解　説	現在ドライバ内に登録されているＰＣＭの、指定ノート番号に登録されているデータを再生します。 入力のD2. Lは、上位１６ビットがノート番号（0～127）、ビット８～１５がサンプリング周波数（0～4）下位８ビットが音声出力モード（1～3）です。

サンプリング周波数
　0 ………… ３．９ＫＨｚ
　1 ………… ５．２ＫＨｚ
　2 ………… ７．８ＫＨｚ
　3 ………… 10．４ＫＨｚ
　4 ………… 15．６ＫＨｚ

音声出力モードの設定内容は、次の通りです。
音声出力モード（オーディオ出力端子）
　1 ………… 左
　2 ………… 右
　3 ………… ステレオ（左右両方）

M_CTRLRES

●ＯＰＭコール番号　：　$3D　ＯＰＭコマンド　：　無し

機　能	指定チャンネルへ各種のオフ出力
入　力	D0.L　$F0　ＩＯＣＳコール番号 D1.L　$3D　ＯＰＭコール番号 D2.W　チャンネル番号（1～25）
リターン	D0.L　エラーコード D0.L＝0………正常終了
解　説	指定のチャンネルにピッチベンドオフ／モジュレーションオフ／ダンパーオフ／オールノートオフを出力します。 入力のD2.Wはチャンネル番号で-1を指定した場合は、全てのチャンネルへピッチベンドオフ／モジュレーションオフ／ダンパーオフ／オールノートオフを出力します。 リターンのD0.Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_OPMREG

●ＯＰＭコール番号　：　$3E　　ＯＰＭコマンド　：　無し

機　能

指定のＯＰＭレジスタに値をセット

入　力

D0. L　　$F0　　ＩＯＣＳコール番号
D1. L　　$3E　　ＯＰＭコール番号
D2. W　　レジスタ番号（0～255）
D3. W　　ＯＰＭへ書き込むデータ（0～255）

リターン

D3. W＝-1　　：
　D0. L　現在のレジスタ値
　　D0. L＝-1　書き換えられていない
D2. W＝-1　　：
　D0. L　ＯＰＭレジスタバッファの先頭アドレス
上記以外　　：
　D0. L　　エラーコード
　　D0. L＝0……正常終了

解　説

指定のＯＰＭレジスタに値をセットします。
入力のD3. Wに－１を指定した場合、指定したＯＰＭレジスタの値を返します。
一度も指定のＯＰＭレジスタが書き換えられていない場合、－１を返します。
D2. Wに－１を指定した場合、ＯＰＭレジスタの先頭アドレスを返します。
ＯＰＭレジスタバッファは各レジスタにつき２バイトずつ計５１２バイト有り、現在までにレジスタが書き換えられてなかった場合は、そのレジスタの位置に－１（ワード）がセットされています。

M_MDREG

●ＯＰＭコール番号　：　$3Ｆ　ＯＰＭコマンド　：　無し

機　能　ＭＩＤＩ制御ＩＣのレジスタに値をセット

入　力

D0.L	$Ｆ0	ＩＯＣＳコール番号
D2.L	グループ番号×65536＋アドレス番号	

リターン

D0.L　エラーコード

D0.L＝0……正常終了

解　説　ＭＩＤＩ制御ＩＣ（ＹM3802）のレジスタに値を書き込みます。
入力のD2.Lは、上位１６ビットがグループ番号（0～9）、下位ビットがアドレス番号（0～7）です。

リターンのD0.Lのエラーコードについては、「１．１３　エラーコード一覧」を参照してください。

注：間違ったパラメータを指定すると、システムが正常に動作しなくなる場合があります。
ＹM3802の機能を十分に理解の上でご使用ください。

M_TNMSET

●ＯＰＭコール番号　：　$40　ＯＰＭコマンド　：　(/V n,name)

機　能　ＦＭ音源の音色名の登録

入　力

D0.L	$Ｆ0	ＩＯＣＳコール番号
D1.L	$40	ＯＰＭコール番号
D2.W	音色番号（1～200）	
A1.L	音色名の先頭アドレス	

リターン

D0.L　エラーコード

D0.L＝0……正常終了

解　説　ＦＭ音源の音色番号で示される音色データに音色名を登録します。
（半角で１０文字以内）
リターンのD0.Lのエラーコードについては、「１．１３　エラーコード一覧」を参照してください。

M_TNMGET

●ＯＰＭコール番号　:　$41　　ＯＰＭコマンド　:　無し

機　能　ＦＭ音源の音色名の取得

入　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$41	ＯＰＭコール番号
D2. W	音色番号（1～200）	
A1. L	音色名を格納するためのバッファ先頭アドレス	

リターン

D0. L　　エラーコード
　D0. L =0 ……… 正常終了
（A1）　　音色名

解　説　ＦＭ音源の音色番号で示される音色データに登録されている音色名を取得します（半角で１０文字以内）。
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PNMSET

●ＯＰＭコール番号　:　$42　　ＯＰＭコマンド　:　(/A n,name)

機　能　ＰＣＭ音色名の登録

入　力

D0. L	$F0	ＩＯＣＳコール番号
D1. L	$42	ＯＰＭコール番号
D2. W	ノート番号（0～127）	
A1. L	音色名の先頭アドレス	

リターン

D0. L　　エラーコード
　D0. L =0 ……… 正常終了

解　説　ＰＣＭのノート番号に音色名を登録します。（半角で１０文字以内）
リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PNMGET

●ＯＰＭコール番号　：　$43　　ＯＰＭコマンド　：　無し

機　能	ＰＣＭ音色名の取得
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$43　　ＯＰＭコール番号 D2. W　　ノート番号（0～127） A1. L　　音色名を格納するためのバッファ先頭アドレス
リターン	D0. L　　エラーコード D0. L＝0……正常終了 （A1）　　音色名
解　説	ＰＣＭ音源の音色名を取得します。（半角で１０文字以内） リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_MSTVOL

●ＯＰＭコール番号　：　$44　　ＯＰＭコマンド　：　(/0)～(/127)

機　能	マスターボリュームの設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$44　　ＯＰＭコール番号 D2. W　　マスターボリューム（0～127）
リターン	D2. W＝-1　： D0. L　現在のマスターボリューム D2. W≠-1　： D0. L　　エラーコード D0. L＝0……正常終了
解　説	マスターボリュームを設定します（デフォルト値は127）。 入力のD2. Wに－１を指定した場合は、現在の設定値を返します。 M_INITを実行するまで有効です。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_BEND

●ＯＰＭコール番号　：　$45　　ＯＰＭコマンド　：　無し

機　能	指定チャンネルのピッチベンド設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$45　　ＯＰＭコール番号 D2. L　　チャンネル番号×65536＋ピッチベンドデータ
リターン	D0. L　　エラーコード D0. L＝0……正常終了
解　説	指定のチャンネルにピッチベンドを設定します。 入力のD2. Lは、上位１６ビットがチャンネル番号（1～25）、 下位１６ビットがピッチベンドデータ（0～127）です。 チャンネル番号に－１を指定した場合は、全てのチャンネルに同じピッチベンドデータを設定します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_MOD

●ＯＰＭコール番号　：　$46　　ＯＰＭコマンド　：　無し

機　能	指定チャンネルのモジュレーション設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$46　　ＯＰＭコール番号 D2. L　　チャンネル番号×65536＋モジュレーションデータ
リターン	D0. L　　エラーコード D0. L＝0……正常終了
解　説	指定のチャンネルのモジュレーションを設定します。 入力のD2. Lは、上位１６ビットがチャンネル番号（1～25）、 下位１６ビットがモジュレーションデータ（0～127）です。 チャンネル番号に－１を指定した場合は、全てのチャンネルに同じモジュレーションデータを設定します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_OPMEXC

●ＯＰＭコール番号　：　$47　　ＯＰＭコマンド　：　(/En)

機　能	ＦＭ音源のパラメータセットの選択
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$47　　ＯＰＭコール番号 D2. W　　パラメータセットモード
リターン	D2. W=-1　： 　D0. L　現在のパラメータセットモード D2. W≠-1　： 　D0. L　エラーコード 　　D0. L=0 ……… 正常終了
解　説	ＦＭ音源にセットする音色パメータの内、ＬＦＱ／ＰＭＤ／ＡＭＤなどの全体に影響するパラメータを音色切り替え時にセットするかどうかを選択します。 入力のD2. Wの設定内容は次の通りです（デフォルト値は1）。

パラメータセットモード

　0 ……………… セットしない
　1 ……………… セットする
－1 ……………… 現在の設定値を返す

リターンのD0. Lのエラーコードについては、「1．13 エラーコード一覧」を参照してください。

M_OPMLFQ

●ＯＰＭコール番号　：　$48　　ＯＰＭコマンド　：　(/SPn)

機　能	ＦＭ音源のＬＦＯの発振周波数の設定
入　力	D0. L　　$F0　　ＩＯＣＳコール番号 D1. L　　$48　　ＯＰＭコール番号 D2. W　　スピードデータ (0～255)
リターン	D2. W＝-1　： 　D0. L　現在のスピードデータ D2. W≠-1　： 　D0. L　　エラーコード 　　D0. L＝0 ………正常終了
解　説	ＦＭ音源の発振周波数を設定します。 入力のD2. Wはスピードデータで、デフォルト値は１９８です。 －１を指定した場合は、現在の設定値を返します。 リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_PGMFLT

●OPMコール番号　：　$49　　OPMコマンド　：　無し

機　能

MML音色切り替えコマンドの出力選択

入　力

D0. L	$F0	IOCSコール番号
D1. L	$49	OPMコール番号
D2. L	チャンネル番号×65536＋フィルタモード	

リターン

D2. W＝-1　：
　D0. L　現在のフィルタモード
D2. W≠-1　：
　D0. L　エラーコード
　　D0. L＝0 ……… 正常終了

解　説

MMLデータ内の音色切り替えコマンド(@n)を、指定チャンネルの演奏時に出力するかどうかを選択します。
入力のD2. Lは、上位16ビットにチャンネル番号（1～25）、下位16ビットが音色切り替えコマンドのフィルタモードです。
チャンネル番号に－1を指定した場合は、全てのチャンネルに同じフィルタモードを設定します。

フィルタモードの設定内容は次の通りです。

　0 ……………… フィルタオフ（@nを出力する）
　1 ……………… フィルタオン（@nを出力しない）
－1 ……………… 現在の設定値を返す

リターンのD0. Lのエラーコードについては、「1．13 エラーコード一覧」を参照してください。

M_VOLFLT

●OPMコール番号　：　$4A　OPMコマンド　：　無し

機　能	MMLボリューム設定コマンドの出力選択
入　力	D0.L　　$F0　　IOCSコール番号 D1.L　　$4A　　OPMコール番号 D2.L　　チャンネル番号×65536＋フィルタモード
リターン	D2.W＝-1　： 　D0.L　現在のフィルタモード D2.W≠-1　： 　D0.L　　エラーコード 　　D0.L＝0 ……… 正常終了
解　説	MMLデータ内のボリューム設定コマンド（Vn/@Vn）を、指定チャンネルの演奏時に出力するかどうかを選択します。 入力のD2.Lは、上位16ビットにチャンネル番号（1～25）、下位16ビットがボリューム設定コマンドのフィルタモードです。 チャンネル番号に－1を指定した場合は、全てのチャンネルに同じフィルタモードを設定します。

フィルタモードの設定内容は次の通りです。

　0 ……………… フィルタオフ（Vn/@Vnを出力する）
　1 ……………… フィルタオン（Vn/@Vnを出力しない）
－1 ……………… 現在の設定値を返す

リターンのD0.Lのエラーコードについては、「1．13 エラーコード一覧」を参照してください。

M_PANFLT

●ＯＰＭコール番号　：　$4Ｂ　　ＯＰＭコマンド　：　無し

機　　能

パンポット設定コマンドの出力選択

入　　力

D0. L　　　$F0　　ＩＯＣＳコール番号
D1. L　　　$4Ｂ　　ＯＰＭコール番号
D2. L　　　チャンネル番号×65536＋フィルタモード

リターン

D2. W＝-1　　：
　　D0. L　現在のフィルタモード
D2. W≠-1　　：
　　D0. L　　　エラーコード
　　　D0. L＝0 ……… 正常終了

解　　説

ＭＭＬデータ内のパンポット設定コマンド（Pn/@Pn）を、指定チャンネルの演奏時に出力するかどうかを選択します。
入力のD2. Lは、上位１６ビットにチャンネル番号（1～25）、
下位１６ビットがパンポット設定コマンドのフィルタモードです。
チャンネル番号に－１を指定した場合は、全てのチャンネルに同じフィルタモードを設定します。

フィルタモードの設定内容は次の通りです。
　　０ ……………… フィルタオフ（Pn/@Pnを出力する）
　　１ ……………… フィルタオン（Pn/@Pnを出力しない）
　－１ ……………… 現在の設定値を返す

リターンのD0. Lのエラーコードについては、「１．１３ エラーコード一覧」を参照してください。

M_VERSION

●ＯＰＭコール番号　：　$5Ｅ　ＯＰＭコマンド　：　無し

機　能	ＯＰＭＤＲＶ３．Ｘバージョンと作成日付の取得
入　力	D0. L　　$Ｆ0　　ＩＯＣＳコール番号 D1. L　　$5Ｅ　　ＯＰＭコール番号
リターン	D0. L　　バージョンと作成年月日
解　説	ＯＰＭＤＲＶ3. Ｘのバージョンと作成年月日を返します。

D31 D30 D29 D28 D27 D26 D25 D24	D23 D22 D21 D20 D19 D18 D17 D16
バージョン（ＢＣＤ）	年（ＢＣＤ）

D15 D14 D13 D12 D11 D10 D09 D08	D07 D06 D05 D04 D03 D02 D01 D00
月（ＢＣＤ）	日（ＢＣＤ）

例）　バージョン１．０　　１９９２年3月3日作成
　　　＄１０９２０３０３

1.10 OPMコマンド

以下のＯＰＭコマンドは、ＯＳ上からデバイス名'OPM'または'MIDI'に直接出力して実行する事ができます。

具体的には、ＭＭＬと共にアスキーファイルとして作成しておき、

```
COPY <ファイル名> OPM⏎

COPY <ファイル名> MIDI⏎
```

で実行できます。

ＯＰＭコマンド	パラメータ　／　機　能
(REM)または /	"(REM)"または"/"で始まる行を注釈行とみなします。 （全角文字で始まる行も注釈とみなします。）
(/OPM) (/OMP) (/MOP) (/MPO) (/PMO) (/POM)	各音源に対するチャンネル番号の割り当てを変更します。 ドライバ組み込み時のオプションスイッチ("/OPM","/POM"など)と同じ機能です。 :OPM =ch1～8　PCM =ch9　MIDI=ch10～25 :OPM =ch1～8　MIDI=ch9～24　PCM =ch25 :MIDI=ch1～16　OPM =ch17～24　PCM =ch25 :MIDI=ch1～16　PCM =ch17　OPM =ch18～25 :PCM =ch1　MIDI=ch2～17　OPM =ch18～25 :PCM =ch1　OPM =ch2～9　MIDI=ch10～25 (ch1～25はチャンネル1～チャンネル25を示します。)
(/Yn)	MUSIC PRO-68K[MIDI]で作成された'～.MUS'ファイルを演奏する場合は(/Y0)を指定してください。その他の場合は(/Y1)を指定してください。ﾃﾞﾌｫﾙﾄは(/Y1)に設定されています。
(/Pn,filename)	:n=ノート番号(0～127) filenameで指定されたＰＣＭデータのファイルを読み込み、n番のノート番号に登録します。ファイル名はフルパス名で指定してください。ファイル名はそのままPCMの音色名として登録されます。
(/An,tonename)	:n=ノート番号(0～127) n番のノート番号にPCMの音色名を登録します。
(/S,filename)	SOUND PRO-68Kの'～.SND'ファイル（または(/SV,filename,n)で作成されたファイル）を読み込み、ＦＭ音源に登録します。 ファイル名はフルパス名で指定してください。 拡張子'.SND'は不要です。ＦＭ音源の音色名も同時に登録します。

ＯＰＭコマンド	パラメータ　／　機　能
(/SV,filename,n1,n2..n3,n4-n5)	:n=音色番号(1～200) SOUND PRO-68Kの '～.SND'ファイルのフォーマットで現在登録されている音色データの内、nで指定された音色データをファイルに保存します。ファイル名はフルパス名で指定してください。 ファイル名に拡張子'.SND'は不要です。 nを省略した場合は200音色全てファイルに登録します。 例)　(/SV,TEST1,1-8,20-30,180-) 音色番号1～8,20～30,180～200 の音色データをTEST1.SNDのファイル名でファイルに保存します。
(/Vn,tonename)	:n=1～200 ＦＭ音源の音色名を登録します。
(/SPn)	:n=0～255 (ﾃﾞﾌｫﾙﾄは198) ＦＭ音源のＬＦＯ（モジュレーション）の変化する速さを設定します。（Ｉ n)でデフォルトに設定されます。
(/En)	:n=0 off　n=1 on (ﾃﾞﾌｫﾙﾄは1) ＦＭ音源に設定する音色パラメータの内、ＬＦＱ／ＰＭＤ／ＡＭＤなどの全体に影響するパラメータを音色切り替え時にセットするかどうかを選択します。 全てのチャンネルに対してモジュレーションを正常に機能させる為には、nに0を指定しておいてください。
(/n)	:n=0～127 (ﾃﾞﾌｫﾙﾄは127) マスターボリュームを設定します。(I n)を実行するまで有効です。 ＰＣＭ音に対しては無効です。
(/PI)	ＰＣＭバッファをクリアします。
(I n)	:n=0 or 1 トラックデータを初期化します。(M t,s)も初期化されます。 各トラックバッファは４バイトに設定されます。 n=0または省略した場合は音色データを初期化しません。 n=1で音色データも初期化します。
(V n1,n2,d1,d2....dm..d55-n2)	:n1=音色番号 (1～200) :n2=書き換えを始めるパラメータ番号 (0～54) :dm=音色パラメータ (0～255) FM音源の音色を設定します。 n2はパラメータの何番から書き換えるかを示しています。
(D n1,n2)	:n1=コピー元の音色番号 (1～200) :n2=コピー先の音色番号 (1～200) ＦＭ音源のn1番の音色をn2番にコピーします。

ＯＰＭコマンド	パラメータ　／　機　能
(M t,s)	:t=トラック番号　s=バッファサイズ トラックバッファを確保します。 tを省略すると全てのトラックにsを設定します。 (I)でデフォルト(全トラック４バイト)に設定されます。 確保したバッファ容量が不足していると、正常に演奏されない場合があります。 (M,T,S)コマンドでエラーが起きた場合は、その時点での確保可能な最大値を設定します。
(A c,t)	:c=チャンネル番号　t=トラック番号 チャンネルcで演奏するトラック番号を指定します。 cが重複した場合は後から指定したﾁｬﾝﾈﾙに設定されます。 tが重複した場合は複数のﾁｬﾝﾈﾙで同じトラックが演奏されます。
(O n)	:n=テンポデータ(20～300)　テンポを設定します。
(Q n1,n2)	:n1=分子　n2=分母　　拍子を設定します。
(T n) MMLdata...	:n=トラック番号(1～80) 指定のトラックにＭＭＬデータを書き込みます。
(E n1,n2,..n25)	:n=チャンネル番号(1～25) nで指定されたチャンネルの音声出力を許可します。nで指定されたチャンネル以外のチャンネルは音声出力が禁止されています。 （内部的な演奏は継続しています） nがすべて省略された場合はすべてのチャンネルの音声出力が許可されます。
(P n1,n2,..n25)	:n=チャンネル番号 nで指定されたチャンネルの演奏を開始します。 nがすべて省略された場合、全てのチャンネルの演奏を開始します。
(S n1,n2,..n25)	:n=チャンネル番号 nで指定されたチャンネルの演奏を停止します。 nがすべて省略された場合、全てのチャンネルの演奏を停止します。
(C n1,n2,..n25)	:n=チャンネル番号 nで指定されたチャンネルの演奏を再開します。 nがすべて省略された場合、全てのチャンネルの演奏を再開します
(W n1,n2,..n25)	:n=チャンネル番号 nで指定されたチャンネルが終了するまで待ちます。(ESCキーまたはBREAKキーで中止します。) nがすべて省略された場合、全てのチャンネルの演奏が終了するまで待ちます。

ＯＰＭコマンド	パラメータ　／　機　　能
(PM n)	:n=小節番号 演奏を開始する小節番号を指定します。 (S n)または演奏終了で解除されます。 (Q n1,n2)で曲の拍子が設定されていなければ正常な演奏ができません。 また、全トラックにおいて、ＭＭＬデータ,"|:",":|","[D.S.]","[D.C.]","[$]"などの数が同じでなければなりません。
(SM n)	:n=小節番号 演奏を停止する小節番号を指定します。 (S n)または演奏終了で解除されます。 (Q n1,n2)で曲の拍子が設定されていなければ正常な演奏ができません。 また、全トラックにおいて、ＭＭＬデータ,"|:"," :|","[D.S.]","[D.C.]","[$]"などの数が同じでなければなりません。
(N c,n)	:c=チャンネル番号(1～25) n=出力チャンネル番号(1～25) 一時的に出力チャンネルを変更します。 (I n)で元に戻ります。
(G c,n)	:c=チャンネル番号 n=音色番号(OPM:1～200 MIDI:1～128) 指定のチャンネルに音色番号を設定します。
(L c,n)	:c=チャンネル番号 n=ボリュームデータ(0～127) 指定のチャンネルにボリュームデータを設定します。
(B c,n)	:c=チャンネル番号 n=パンポットデータ(0～127) 指定のチャンネルにパンポットデータを設定します。
(U c,n)	:c=チャンネル番号 n=ベロシティデータ(0～127) 指定のチャンネルにベロシティデータを設定します。
(F c,n)	:c=チャンネル番号 n=トランスポーズデータ(0～48) 指定のチャンネルにトランスポーズデータを設定します。 ±２ｵｸﾀｰﾌﾞの範囲で半音ずつ音程をシフトします。(ﾃﾞﾌｫﾙﾄは24)
(Y c,n1,n2)	:c=チャンネル番号 n1=コントロールコード n2=データ 指定のチャンネルにコントロールデータを出力します。
(J c)	:c=チャンネル番号(1～25) 指定のチャンネルへオールノートオフを出力します cを省略した場合は全てのチャンネルにオールノートオフを出力します。
(X n1,n2,.... n512)	n1～nmまでのデータをＭＩＤＩへ直接出力します。 指定できるデータの数は５１２個まででです。
(Z n)	:n=0 内部同期　n=1 外部同期　n=2 ＦＳＫ同期
(P) (S) (C)	ＭＩＤＩのスタートコード($FA)を出力 ＭＩＤＩのストップコード($FC)を出力 ＭＩＤＩのコンティニューコード($FB)を出力 常にＭＩＤＩクロック($F8)を出力しているので、外部シーケンサーと同期演奏させることができます。

1.13 エラーコード一覧

エラー番号	内容
1	テンポの指定が無効です
2	トラック番号が無効です
3	OPMDRV3.X が登録されていません
4	サイズの指定が無効です
5	メモリーの確保ができません
6	チャンネル番号が無効です
7	配列の指定に誤りがあります
8	OPMDRV3.X は現在停止中です
9	ベロシティの値がありません
10	ベロシティの値が無効です
11	録音モードの指定が無効です
12	録音の準備ができていません
13	パンポットの値がありません
14	パンポットの値が無効です
15	トランスポーズの値がありません
16	トランスポーズの値が無効です
17	パラメータを両方省略する事はできません
18	パラメータの指定に誤りがあります
19	ＭＭＬの文法に誤りがあります
20	［］の中の指定に誤りがあります
21	］がありません
22	繰り返しの値が無効です
23	繰り返し番号の指定がありません
24	繰り返し番号の指定が無効です
25	オクターブの番号がありません
26	オクターブの指定が無効です
27	長さの値が無効です
28	トラック容量が足りません
29	音色番号の指定がありません
30	＠Ｗの値がありません
31	＠Ｗの値が無効です
32	テンポの値がありません
33	テンポの値が無効です
34	長さの値がありません
35	音量の値がありません

エラー番号	内　　　　　容
３６	音量の値が無効です
３７	キーコードの値がありません
３８	キーコードの値が無効です
３９	音色番号が無効です
４０	｝がありません
４１	｛｝の中に音符が多すぎます
４２	｛｝の中に音符がありません
４３	Ｑの指定に誤りがあります
４４	｛｝の中の文法に誤りがあります
４５	Ｙの指定に誤りがあります
４６	@Ｌの指定に誤りがあります
４７	Ｐの指定に誤りがあります
４８	タイの指定に誤りがあります
４９	和音の指定に誤りがあります
５０	和音の数が多すぎるか、または 後の ’ がありません
５１	@Ｖの指定に誤りがあります
５２	@Ｐの指定に誤りがあります
５３	@Ｕの指定に誤りがあります
５４	@Ｎの指定に誤りがあります
５５	@Ｍの指定に誤りがあります
５６	@Ｄの指定に誤りがあります
５７	@Ｂの指定に誤りがあります
５６	@Ｔの指定に誤りがあります
５９	@Ｙの指定に誤りがあります
６０	ＰＣＭのバッファ不足で、これ以上登録できません
６１	ＰＣＭは使用できません
６２	OPMDRV3.X のバージョンが違います
６３	無効なサンプリング周波数を指定しました
６４	無効な出力モードを指定しました
６５	無効なデータサイズを指定しました

第2章　ＳＣＳＩ

「Ｃ　ｃｏｍｐｉｌｅｒ　ＰＲＯ－６８Ｋ　ｖｅｒ２．１」では新たにＳＣＳＩに対応したライブラリを用意しています。

これにより、以下のファイルが変更／追加されています。

ファイル名	内　　容	ディレクトリ
IOCSLIB.L	ＩＯＣＳコールライブラリ	ｼｽﾃﾑ2の¥LIB
IOCSLIB.H	ＩＯＣＳコール関数の定義	ｼｽﾃﾑ2の¥INCLUDE
IOCSCALL.MAC	ＩＯＣＳコールのヘッダファイル	ｼｽﾃﾑ2の¥INCLUDE
SCSI1.C	ＳＣＳＩサンプル１	ｼｽﾃﾑ2の¥SAMPLE
SCSI2.C	ＳＣＳＩサンプル２	ｼｽﾃﾑ2の¥SAMPLE

2.1 SCSIサンプルプログラム

ＳＣＳＩライブラリを用いたサンプルプログラムとして、"SCSI1.C"および"SCSI2.C"がシステムディスク２のＳＡＭＰＩＬＥディレクトリに入っています。

1.　ＳＣＳＩ１．Ｃ

ＳＣＳＩコール番号＄２０以上の高レベルなライブラリを用いたサンプルプログラムです。ＳＣＳＩデータのダンプ表示やＳＣＳＩ装置情報の表示などを行っています。

```
A>CC /Y SCSI1.C
```

としてコンパイルしてください。引数にＩＤのみを指定した（引数が１つの）ときは、ＳＣＳＩ装置情報を表示し、ブロック番号も指定したときはそのブロック番号のデータを表示します。その後は以下のキー操作をおこなえます。

ROLLUP,RETURN	：	次のブロック番号のデータを表示
ROLLDOWUN,UNDO	：	前のブロック番号のデータを表示
ESC	：	終了

例１）ＳＣＳＩ装置のＩＤ番号０の装置情報を表示。

```
A>SCSI1 0
SCSI Library sample 1 version 1.00 by SHARP
ＩＤ番号　　　：0
最大ブロック数：167720
ブロックサイズ：512バイト
最大容量　　　：81Mバイト
周辺装置機種　：直接アクセス装置
媒体の取外し　：不可
ＡＮＳＩ規格　：合致
A>
```

例2）ＳＣＳＩ装置のＩＤ番号０、ブロック番号４のデータを表示。

```
  A>SCSI1 0 4
SCSI Library sample 1 version 1.00 by SHARP
[ID=0] [BLOCK=4] [MAX BLOCK=167719] [BLOCK SIZE=512] [最大容量=81Mbyte]
BLOCK+n   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII charactor
00000000 58 36 38 4B 00 01 40 20 00 01 47 93 00 01 47 93 X68K..@ ..G ..G滴
00000010 48 75 6D 61 6E 36 38 6B 00 00 00 20 00 00 A0 00 Human68k... ... .
00000020 48 75 6D 61 6E 36 38 6B 02 00 A0 20 00 00 A0 00 Human68k..  ... .
～                                                                         ～
00000100 91 95 92 75 91 53 91 CC 82 CC 97 65 97 CA 20 20 装置全体の容量
00000110 20 20 20 20 20 20 20 20 20 20 38 31 20 82 6C 83            81 Mバ
00000120 6F 83 43 83 67 00 00 00 98 41 91 B1 8A 6D 95 DB oイト...連続確保
00000130 89 C2 94 5C 82 C8 8D C5 91 E5 97 65 97 CA 20 20 可能な最大容量
00000140 20 20 38 31 20 82 6C 83 6F 83 43 83 67 00 00 00   81 Mバイト...
～                                                                         ～
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
  キー入力待ち。
```

２．　ＳＣＳＩ２．Ｃ

ＳＣＳＩコール番号＄２０未満の低レベルなライブラリを用いたサンプルプログラムです。Ｓ＿ＲＥＡＤライブラリと同様にＳＣＳＩ装置からデータを読み込みます。

```
A>CC /Y SCSI1.C SCSI2.C
```

としてコンパイルしてください。"SCSI1.C"のＳ＿ＲＥＡＤ関数には、"SCSI2.C"のプログラムが、ＳＣＳＩライブラリ内のＳ＿ＲＥＡＤライブラリよりも優先して使用されます。

2.3 ライブラリ一覧

● ライブラリ一覧（ＳＣＳＩコール番号順）

SCSI番号	ﾗｲﾌﾞﾗﾘ関数名	機　　能	PAGE
$00	S_RESET	SPC(SCSIプロトコルコントローラ)およびSCSIバスのリセット	152
$01	S_SELECT	アービトレーションフェーズとセレクションフェーズの実行	154
$03	S_CMDOUT	コマンドアウトフェーズの実行	146
$04	S_DATAIN	データインフェーズの実行	146
$05	S_DATAOUT	データアウトフェーズの実行	147
$06	S_STSIN	ステータスインフェーズの実行	155
$07	S_MSGIN	メッセージインフェーズの実行	148
$08	S_MSGOUT	メッセージアウトフェーズの実行	149
$09	S_PHASE	フェーズセンス	150
$20	S_INQUIRY	ＩＮＱＵＩＲＹデータの要求	148
$21	S_READ	ＳＣＳＩ装置よりデータの読み込み	150
$22	S_WRITE	ＳＣＳＩ装置へのデータの書き込み	156
$23	S_FORMAT	ＳＣＳＩ装置のフォーマット	147
$24	S_TESTUNIT	ＳＣＳＩ装置が動作可能であるかどうかの調査	156
$25	S_READCAP	ＳＣＳＩ装置の容量に関する情報の調査	151
$26	S_READEXT	拡張ＲＥＡＤコマンド	151
$27	S_WRITEEXT	拡張ＷＲＩＴＥコマンド	157
$2b	S_REZEROUNIT	ＳＣＳＩ装置を指定の状態にセット	153
$2c	S_REQUEST	ＳＣＳＩ装置のセンスデータの調査	153
$2d	S_SEEK	指定の論理ブロックアドレスへのシーク	154
$2f	S_STARTSTOP	ＳＣＳＩ装置に対して以降の操作の可能／不可能の要求	155
$31	S_REASSIGN	欠陥ブロックの割り当ての要求	152
$32	S_PAMEDIUM	メディアのイジェクトの禁止・許可の設定	149

2．4 ライブラリ リファレンス

「C compiler PRO－68K ver2．1」では、SCSIドライバに対応したライブラリがサポートされています。
これにより、新しく次のファイルがver2．1より追加／変更されています。

ファイル名	内　容
IOCSLIB.L	ＩＯＣＳコールライブラリ
IOCSLIB.H	ＩＯＣＳコール関数の定
IOCSCALL.MAC	ＩＯＣＳコールのヘッダファイル

S_CMDOUT

レベル0

書式

```
#include <iocslib.h>

int S_CMDOUT(BYTE,CDBADDR);
int BYTE;                          /*コマンドのバイト数*/
unsigned char *CDBADDR;            /*CDBの先頭アドレス*/
```

機能

コマンドフェーズを実行します。
CDB(Command Discriptor Block)の先頭アドレスからそのコマンドのバイト数分のデータをＳＣＳＩバス上に出力します。
グループ0/1/5以外のコマンドのときは、BYTEにコマンドのバイト数を指定してください。

戻り値

エラーコードを返します。（エラーコード参照）

S_DATAIN

レベル0

書式

```
#include <iocslib.h>

int S_DATAIN(BYTE,ADDRESS);
int BYTE;                          /*読み込みバイト数*/
unsigned char *ADDRESS;            /*読み込み先頭アドレス*/
```

機能

データインフェーズを実行します。
ADDRESSで指定したアドレスへBYTEで指定したバイト数分のデータをＳＣＳＩバス上から読み込みます。

戻り値

エラーコードを返します。（エラーコード参照）

S_DATAOUT

レベル0

書式

```
#include <iocslib.h>

int S_DATAOUT(BYTE,ADDRESS);
int BYTE;                          /*書き込みバイト数*/
unsigned char *ADDRESS;            /*書き込み先頭アドレス*/
```

機能

データアウトフェーズを実行します。
ADDRESSで指定したアドレスからBYTEで指定したバイト数分のデータをＳＣＳＩバス上へ書き込みます。

戻り値

エラーコードを返します。（エラーコード参照）

S_FORMAT

レベル0

書式

```
#include <iocslib.h>

int S_FORMAT(INTERL,ID);
int INTERL;                        /*インターリーブ*/
int ID;                            /*ターゲットＩＤ*/
```

機能

指定のインターリーブ値でＳＣＳＩ装置をフォーマットします。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_INQUIRY

レベル０

書式

```
#include <iocslib.h>

int S_INQUIRY(BYTE,ID,inq);
int BYTE;                        /*読み込みバイト数*/
int ID;                          /*ターゲットのＩＤ*/
struct INQUIRY *inq;             /*読み込み先頭アドレス*/
```

機能

inqで示すアドレスにBYTEで指定したバイトのINQUIRYデータを読み込みます。

```
struct INQUIRY {
        unsigned char unit;      /*周辺装置機種*/
        unsigned char info;      /*RMB|機種限定*/
        unsigned char ver;       /*ISO|ECMA|ANSIバージョン*/
        unsigned char reserve;   /*保留*/
        unsigned char size;      /*追加長*/
        unsigned char buff[];    /*ベンダーユニーク*/
};
```

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_MSGIN

レベル０

書式

```
#include <iocslib.h>

int S_MSGIN(ADDRESS);
unsigned char *ADDRESS;          /*読み込み先頭アドレス*/
```

機能

メッセージインフェーズを実行します。
ADDRESSで指定したアドレスへ１バイトのデータをＳＣＳＩバス上から読み込みます。

戻り値

エラーコードを返します。（エラーコード参照）

S_MSGOUT

レベル０

書式

```
#include <iocslib.h>

int S_MSGOUT(ADDRESS);
unsigned char *ADDRESS;          /*読み込み先頭アドレス*/
```

機能

メッセージアウトフェーズを実行します。
ADDRESSで指定したアドレスから１バイトのデータをＳＣＳＩバス上へ書き込みます。

戻り値

エラーコードを返します。（エラーコード参照）

S_PAMEDIUM

レベル０

書式

```
#include <iocslib.h>

int S_PAMEDIUM(MODE,ID);
int MODE;              /*0:イジェクト許可／1:イジェクト禁止*/
int ID;                /*ターゲットのＩＤ*/
```

機能

メディアのイジェクトの禁止／許可を設定します。MODEが０のときイジェクト許可、１のときイジェクト禁止に設定します。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_PHASE

レベル０

書式

```
#include <iocslib.h>

int S_PHASE(void);
```

機能

フェーズをセンスします。
SPCのPSNSレジスタを読みだします。

戻り値

現在のフェーズを返します。

S_READ

レベル０

書式

```
#include <iocslib.h>

int S_READ(POS,BLOCK,ID,SIZE,ADDRESS);
int POS;               /*目的読み込み位置（論理ブロック番号）*/
int BLOCK;             /*読み込みブロック数（論理ブロック数）*/
int ID;                /*ターゲットのＩＤ*/
int SIZE;              /*１ブロックの容量（0=256/1=512/2=1024）*/
unsigned char *ADDRESS;
                       /*読み込み先頭アドレス*/
```

機能

POSで指定した論理ブロック番号からBLOCKで指定したブロック分のデータをADDRESSで指定したアドレスへ読み込みます。
１ブロックの容量は３以上を指定できません。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１，－２なら異常終了

S_READCAP

レベル０

書式

```
#include <iocslib.h>

int S_READCAP(ID,capa);
int ID;                         /*ターゲットＩＤ*/
struct READCAP *capa            /*読み込み先頭アドレス*/
```

機能

capaで示すアドレスに8バイトのREAD CAPACITYデータを読み込みます。

```
struct READCAPA {
        unsigned int block;     /*論理ブロックアドレス*/
        unsigned int size;      /*ブロック長*/
};
```

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_READEXT

レベル０

書式

```
#include <iocslib.h>

int S_READEXT(POS,BLOCK,ID,SIZE,ADDRESS);
int POS;              /*目的読み込み位置（論理ブロック番号）*/
int BLOCK;            /*読み込みブロック数（論理ブロック数）*/
int ID;               /*ターゲットのＩＤ*/
int SIZE;             /*１ブロックの容量（0=256/1=512/2=1024）*/
unsigned char *ADDRESS;
                      /*読み込み先頭アドレス*/
```

機能

POSで指定した論理ブロック番号からBLOCKで指定したブロック分のデータをADDRESSで指定したアドレスへ読み込みます。
グループ１のコマンド。
読み込みブロック長は65,535ブロックまで指定可能。
１ブロックの容量は３以上を指定できません。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１，－２なら異常終了

S_REASSIGN

レベル０

書式

```
#include <iocslib.h>

int S_REASSIGN(BYTE,ID,ADDRESS);
int BYTE;                        /*書き込みバイト数*/
int ID;                          /*ターゲットのＩＤ*/
unsigned char *ADDRESS;          /*書き込み先頭アドレス*/
```

機能

欠陥ブロックの再割り当てを要求します。ADDRESSで指定したアドレスからBYTEで指定したバイト数のREASSIGN BLOCKSデータを書き込みます。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_RESET

レベル０

書式

```
#include <iocslib.h>

void S_RESET(void);
```

機能

ＳＰＣ（ＳＣＳＩプロトコルコントローラ）の初期化、ＳＣＳＩバスのリセットを行います。
ＳＣＳＩバスリセット後に２秒間待機します。

戻り値

戻り値はありません。

S_REQUEST

レベル0

書式

```
#include <iocslib.h>

int S_REQUEST(BYTE,ID,ADDRESS);
int BYTE;                           /*読み込みバイト数*/
int ID;                             /*ターゲットのＩＤ*/
unsigned char *ADDRESS;             /*読み込み先頭アドレス*/
```

機能　ADDRESSで指定したアドレスへBYTEで指定したバイト数のREQUEST SENSEデータ（ＳＣＳＩ装置のセンスデータ）を読み込みます。

戻り値　ステータスを返します。（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_REZEROUNIT

レベル0

書式

```
#include <iocslib.h>

int S_REZEROUNIT(ID);
int ID;                             /*ターゲットのＩＤ*/
```

機能　ＳＣＳＩ装置を指定の状態にセットすることを要求します。
指定の状態は各装置の取扱説明書を参照してください。

戻り値　ステータスを返します。（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_SEEK

レベル０

書式

```
#include <iocslib.h>

int S_SEEK(POS,ID);
int POS;                /*目的シーク位置（論理ブロック番号）*/
int ID;                 /*ターゲットのＩＤ*/
```

機能

POSで指定した論理ブロック番号にシークします。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_SELECT

レベル０

書式

```
#include <iocslib.h>

int S_SELECT(ID);
int ID;                          /*ターゲットＩＤ*/
```

機能

アービトレーションフェーズとセレクションフェーズを実行します。

戻り値

エラーコードを返します。（エラーコードの参照）

S_STARTSTOP

レベル０

書式

```
#include <iocslib.h>

int S_STARTSTOP(MODE,ID);
int MODE;                            /*0:操作不可／1:操作可*/
int ID;                              /*ターゲットのＩＤ*/
```

機能

ＳＣＳＩ装置に対して以降の操作を可能、または操作を不可能にすることを要求します。
MODEが１のときはＳＣＳＩ装置を操作できるように設定し、また０のときはＳＣＳＩ装置を停止（操作不可）するように要求します。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_STSIN

レベル０

書式

```
#include <iocslib.h>

int S_STSIN(ADDRESS);
unsigned char *ADDRESS;              /*読み込み先頭アドレス*/
```

機能

ステータスインフェーズを実行します。
ADDRESSで指定したアドレスへ１バイトのデータをＳＣＳＩバス上から読み込みます。

戻り値

エラーコードを返します。（エラーコードの参照）

S_TESTUNIT

レベル０

書式

```
#include <iocslib.h>

int S_TESTUNIT(ID);
int ID;                         /*ターゲットＩＤ*/
```

機能

SCSI装置が動作可能であるかを調べます。戻り値がNULLのときにのみ動作可能です。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１なら異常終了

S_WRITE

レベル０

書式

```
#include <iocslib.h>

int S_WRITE(POS,BLOCK,ID,SIZE,ADDRESS);
int POS;              /*目的書き込み位置（論理ブロック番号）*/
int BLOCK;            /*書き込みブロック数（論理ブロック数）*/
int ID;               /*ターゲットのＩＤ*/
int SIZE;             /*１ブロックの容量（0=256/1=512/2=1024）*/
unsigned char *ADDRESS;
                      /*書き込み先頭アドレス*/
```

機能

POSで指定した論理ブロック番号からBLOCKで指定したブロック分のデータをADDRESSで指定したアドレスから書き込みます。
１ブロックの容量は３以上を指定できません。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１，－２なら異常終了

S_WRITEEXT

レベル０

書式

```
#include <iocslib.h>

int S_WRITEEXT(POS,BLOCK,ID,SIZE,ADDRESS);
int POS;              /*目的書き込み位置（論理ブロック番号）*/
int BLOCK;            /*書き込みブロック数（論理ブロック数）*/
int ID;               /*ターゲットのＩＤ*/
int SIZE;             /*１ブロックの容量（0=256/1=512/2=1024）*/
unsigned char *ADDRESS;
                      /*書き込み先頭アドレス*/
```

機能

POSで指定した論理ブロック番号からBLOCKで指定したブロック分のデータをADDRESSで指定したアドレスへ書き込みます。
グループ１のコマンド。
書き込みブロック長は65,535ブロックまで指定可能。
１ブロックの容量は３以上を指定できません。

戻り値

ステータスを返します。（上位＝メッセージインフェーズで得た情報
／下位＝ステータスインフェーズで得た情報）
－１，－２なら異常終了

2.4 SCSI IOCS一覧

● ＩＯＣＳ一覧（アルファベット順）

SCSI番号	IOCSコール名	機能	PAGE
$03	S_CMDOUT	コマンドアウトフェーズの実行	161
$04	S_DATAIN	データインフェーズの実行	161
$05	S_DATAOUT	データアウトフェーズの実行	162
$23	S_FORMAT	ＳＣＳＩ装置のフォーマット	166
$20	S_INQUIRY	ＩＮＱＵＩＲＹデータの要求	164
$07	S_MSGIN	メッセージインフェーズの実行	163
$08	S_MSGOUT	メッセージアウトフェーズの実行	163
$32	S_PAMEDIUM	メディアのイジェクトの禁止・許可の設定	171
$09	S_PHASE	フェーズセンス	164
$21	S_READ	ＳＣＳＩ装置よりデータの読み込み	165
$25	S_READCAP	ＳＣＳＩ装置の容量に関する情報の調査	167
$26	S_READEXT	拡張ＲＥＡＤコマンド	167
$31	S_REASSIGN	欠陥ブロックの割り当ての要求	170
$00	S_RESET	SPC(SCSIプロトコルコントローラ)およびSCSIバスのリセット	160
$2c	S_REQUEST	ＳＣＳＩ装置のセンスデータの調査	169
$2b	S_REZEROUNIT	ＳＣＳＩ装置を指定の状態にセット	168
$2d	S_SEEK	指定の論理ブロックアドレスへのシーク	169
$01	S_SELECT	アービトレーションフェーズとセレクションフェーズの実行	160
$2f	S_STARTSTOP	ＳＣＳＩ装置に対して以降の操作の可能／不可能の要求	170
$06	S_STSIN	ステータスインフェーズの実行	162
$24	S_TESTUNIT	ＳＣＳＩ装置が動作可能であるかどうかの調査	166
$22	S_WRITE	ＳＣＳＩ装置へのデータの書き込み	165
$27	S_WRITEEXT	拡張ＷＲＩＴＥコマンド	168

2.5 IOCSコールリファレンス

「Ｃ compiler ＰＲＯ－６８Ｋ ver２．１」では、ＳＣＳＩドライバにより、次の様にしてＩＯＣＳコールを利用することができます。

```
moveq.l  #SCSIコール番号, D1
moveq.l  #$F5, D0
trap     #15
```

（ＳＣＳＩのＩＯＣＳコール番号は全て＄Ｆ５です。）

例）　ＳＰＣのリセットおよびＳＣＳＩバスのリセット

```
moveq.l   #$00,D1
moveq.l   #$F5,D0
trap      #15
```

またはマクロを使用して

```
include   iocscall.mac
              :
              :
moveq.l   #_S_RESET
IOCS      _SCSIDRV
```

ＩＯＣＳコールの詳細説明は、「プログラマーズマニュアル第３章ＩＯＣＳコール」を参照してください。

S_RESET

● S_RESET　ＩＯＣＳコール番号：＄Ｆ５　　ＳＣＳＩコール番号：＄００

機　能	ＳＰＣ（ＳＣＳＩプロコトルコントローラ）のリセットおよびＳＣＳＩバスリセット
入　力	D0.L　＄F5　ＩＯＣＳコール番号 D1.L　$00　ＳＣＳＩコール番号
リターン	なし（D0.Lは内容が保証されません）
解　説	ＳＰＣの初期化、ＳＣＳＩバスのリセットを行います。 ＳＣＳＩバスリセット後に２秒間待機します。

S_SELECT

●S_SELECT　ＩＯＣＳコール番号：＄Ｆ５　　ＳＣＳＩコール番号：＄01

機　能	アービトレーションフェーズとセレクションフェーズの実行
入　力	D0.L　＄F5　ＩＯＣＳコール番号 D1.L　$01　ＳＣＳＩコール番号 D4.L　ターゲットのＩＤ
リターン	D0.L　エラーコード（０以外なら異常終了）
解　説	アービトレーションフェーズとセレクションフェーズを実行します。

S_CMDOUT

● S_CMDOUT　ＩＯＣＳコール番号：＄Ｆ５　　ＳＣＳＩコール番号：＄０３

機　能	コマンドアウトフェーズの実行	
入　力	D0. L	＄F5　　ＩＯＣＳコール番号
	D1. L	$03　　ＳＣＳＩコール番号
	D3. L	コマンドのバイト数（グループ0/1/5以外のとき）
	A1. L	ＣＤＢ（ＣＯＭＭＡＮＤ　ＤＩＳＣＲＩＰＴＯＲ ＢＬＯＣＫ）の先頭アドレス
リターン	D0. L	エラーコード

解　説

コマンドフェーズを実行します。
A1. Lのアドレスからそのコマンドのバイト数分のデータをＳＣＳＩバス上に出力します。
グループ0/1/5以外のコマンドのときはD3. Lにバイト数を指定してください。
リターンのD0. L のエラーコードについては、エラーコード表を参照してください。

S_DATAIN

● S_DATAIN　ＩＯＣＳコール番号：＄Ｆ５　　ＳＣＳＩコール番号：＄０４

機　能	データインフェーズの実行	
入　力	D0. L	＄F5　　ＩＯＣＳコール番号
	D1. L	$04　　ＳＣＳＩコール番号
	D3. L	読み込みバイト数
	A1. L	読み込み先頭アドレス
リターン	D0. L	エラーコード

解　説

データインフェーズを実行します。
A1. LのアドレスへD3. LバイトのデータをＳＣＳＩから読み込みます。
リターンのD0. Lのエラーコードについては、エラーコード表を参照してください。

S_DATAOUT

● S_DATAOUT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄０５

機　能	データアウトフェーズの実行		
入　力	D0. L	＄F5	ＩＯＣＳコール番号
	D1. L	$05	ＳＣＳＩコール番号
	D3. L	書き込みバイト数	
	A1. L	書き込み先頭アドレス	
リターン	D0. L	エラーコード	

解　説

データアウトフェーズを実行します。
A1. LのアドレスからD3. LバイトのデータをＳＣＳＩへ書き込みます。
リターンのD0. Lのエラーコードについては、エラーコード表を参照してください。

S_STSIN

● S_STSIN　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄０６

機　能	ステータスインフェーズの実行		
入　力	D0. L	＄F5	ＩＯＣＳコール番号
	D1. L	$06	ＳＣＳＩコール番号
	A1. L	読み込み先頭アドレス	
リターン	D0. L	エラーコード	

解　説

ステータスインフェーズを実行します。
A1. Lのアドレスへ１バイトのデータをＳＣＳＩバスから読み込みます。
リターンのD0. Lのエラーコードについては、エラーコード表を参照してください。

S_MSGIN

● S_MSGIN　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄０７

機　能	メッセージインフェーズの実行		
入　力	D0. L	$F5	ＩＯＣＳコール番号
	D1. L	$07	ＳＣＳＩコール番号
	A1. L	読み込み先頭アドレス	
リターン	D0. L	エラーコード	

解　説

メッセージインフェーズを実行します。
A1. Lのアドレスへ１バイトのデータをＳＣＳＩバス上から読み込みます。
リターンのD0. Lのエラーコードについては、エラーコード表を参照してください。

S_MSGOUT

● S_MSGOUT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄０８

機　能	メッセージアウトフェーズの実行		
入　力	D0. L	$F5	ＩＯＣＳコール番号
	D1. L	$08	ＳＣＳＩコール番号
	A1. L	書き込み先頭アドレス	
リターン	D0. L	エラーコード	

解　説

メッセージアウトフェーズを実行します。
A1. Lのアドレスから１バイトのデータをＳＣＳＩバス上へ書き込みます。
リターンのD0. Lのエラーコードについては、エラーコード表を参照してください。

S_PHASE

● S_PHASE　ＩＯＣＳコール番号：＄Ｆ５　　ＳＣＳＩコール番号：＄０９

機　能	フェーズセンス		
入　力	D0.L	＄F5	ＩＯＣＳコール番号
	D1.L	$09	ＳＣＳＩコール番号
リターン	D0.L	現在のフェーズ	
解　説	ＳＰＣのＰＳＮＳレジスタ（エラーコード表を参照）を読みだします。		

S_INQUIRY

● S_INQUIRY　ＩＯＣＳコール番号：＄Ｆ５　　ＳＣＳＩコール番号：＄２０

機　能	ＩＮＱＵＩＲＹデータの要求		
入　力	D0.L	＄F5	ＩＯＣＳコール番号
	D1.L	$20	ＳＣＳＩコール番号
	D3.L	読み込みバイト数	
	D4.L	ターゲットのＩＤ	
	A1.L	読み込み先頭アドレス	
リターン	D0.L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了	
解　説	A1.LのアドレスへD3.LバイトのＩＮＱＵＩＲＹデータを読み込みます。		

S_READ

● S_READ　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２１

機　能	ＳＣＳＩ装置よりデータの読み込み	
入　力	D0. L	＄F5　ＩＯＣＳコール番号
	D1. L	$21　ＳＣＳＩコール番号
	D2. L	目的読み込み位置（物理ブロック番号）
	D3. L	読み込みブロック数（論理ブロック数）
	D4. L	ターゲットのＩＤ
	D5. L	１ブロックの容量(0=256/1=512/2=1024)
	A1. L	読み込み先頭アドレス
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１，－２なら異常終了
解　説	D2. Lの論理ブロック番号からD3. Lブロック分のデータをA1. Lのｱﾄﾞﾚｽへ読み込みます。１ブロックの容量は３以上を指定できません。	

S_WRITE

● S_WRITE　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２２

機　能	ＳＣＳＩ装置へのデータの書き込み	
入　力	D0. L	＄F5　ＩＯＣＳコール番号
	D1. L	$22　ＳＣＳＩコール番号
	D2. L	目的書き込み位置（論理ブロック番号）
	D3. L	書き込みブロック数（論理ブロック数）
	D4. L	ターゲットのＩＤ
	D5. L	１ブロックの容量(0=256/1=512/2=1024)
	A1. L	書き込み先頭アドレス
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１，－２なら異常終了
解　説	D2. Lの論理ブロック番号からD3. Lブロック分のデータをA1. Lのアドレスから書き込みます。 １ブロックの容量は３以上を指定できません。	

S_FORMAT

● S_FORMAT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２３

機　能	ＳＣＳＩ装置のフォーマット	
入　力	D0. L	＄F5　ＩＯＣＳコール番号
	D1. L	$23　ＳＣＳＩコール番号
	D3. L	インターリーブ
	D4. L	ターゲットのＩＤ
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了
解　説	指定のインターリーブ値でＳＣＳＩ装置をフォーマットします。	

S_TESTUNIT

● S_TESTUNIT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２４

機　能	ＳＣＳＩ装置が動作可能であるかどうかを調べる。	
入　力	D0. L	＄F5　ＩＯＣＳコール番号
	D1. L	$24　ＳＣＳＩコール番号
	D4. L	ターゲットのＩＤ
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了
解　説	ＳＣＳＩ装置が動作可能であるかどうかを調べる。リターンのD0. Lが0のみ動作可能です。	

S_READCAP

● S_READCAP　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２５

機　能	ＳＣＳＩの容量に関する情報を調べる。	
入　力	D0. L	＄F5　ＩＯＣＳコール番号
	D1. L	$25　ＳＣＳＩコール番号
	D4. L	ターゲットのＩＤ
	A1. L	読み込み先頭アドレス
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了
解　説	A1. Lのアドレスへ４ワードのＲＥＡＤ　ＣＡＰＡＣＩＴＹデータを読み込みます。	

S_READEXT

● S_READEXT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２６

機　能	拡張ＲＥＡＤコマンド	
入　力	D0. L	＄F5　ＩＯＣＳコール番号
	D1. L	$26　ＳＣＳＩコール番号
	D2. L	目的読み込み位置（論理ブロック番号）
	D3. L	読み込みブロック数（物理セクタ数）
	D4. L	ターゲットのＩＤ
	D5. L	１ブロックの容量(0=256/1=512/2=1024)
	A1. L	読み込み先頭アドレス
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１，－２なら異常終了
解　説	グループ１のコマンド。読み込みブロック長は65,535ブロックまで指定可能。D2. Lの論理ブロック番号からD3. Lブロック分のデータをA1. Lのアドレスへ読み込みます。 １ブロックの容量は３以上を指定できません。	

S_WRITEEXT

● S_WRITEEXT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２７

機　能	拡張ＷＲＩＴＥコマンド	
入　力	D0. L	$F5　ＩＯＣＳコール番号
	D1. L	$27　ＳＣＳＩコール番号
	D2. L	目的書き込み位置（論理ブロック番号）
	D3. L	書き込みブロック数（論理ブロック数）
	D4. L	ターゲットのＩＤ
	D5. L	１ブロックの容量(0=256/1=512/2=1024)
	A1. L	書き込み先頭アドレス
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１，－２なら異常終了

解　説

グループ１のコマンド。書き込みブロック長は65,535まで指定可能。D2. Lの論理ブロック番号からD3. Lブロック分のデータをA1. Lのアドレスへ書き込みます。

１ブロックの容量 は３以上を指定できません。

S_REZEROUNIT

● S_REZEROUNIT　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２Ｂ

機　能	ＳＣＳＩ装置を指定の状態にセット	
入　力	D0. L	$F5　ＩＯＣＳコール番号
	D1. L	$2B　ＳＣＳＩコール番号
	D4. L	ターゲットのＩＤ
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了

解　説

ＳＣＳＩ装置を指定の状態にセットすることを要求します。
指定の状態は各装置の取扱説明書を参照してください。

S_REQUEST

● S_REQUEST　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２Ｃ

機　能	ＳＣＳＩ装置のセンスデータを調べる	
入　力	D0.L	＄Ｆ５　ＩＯＣＳコール番号
	D1.L	＄２Ｃ　ＳＣＳＩコール番号
	D3.L	読み込みバイト数
	D4.L	ターゲットのＩＤ
	A1.L	読み込み先頭アドレス
リターン	D0.L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了
解　説	A1.LのアドレスへD3.LのバイトЖ数分のＲＥＱＵＥＳＴ ＳＥＮＳＥデータ（ＳＣＳＩ装置のセンスデータ）を読み込みます。	

S_SEEK

● S_SEEK　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２Ｄ

機　能	指定の論理ブロックアドレスへシークする	
入　力	D0.L	＄Ｆ５　ＩＯＣＳコール番号
	D1.L	＄２Ｄ　ＳＣＳＩコール番号
	D2.L	目的シーク位置（論理ブロック番号）
	D4.L	ターゲットのＩＤ
リターン	D0.L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了
解　説	D2.Lの論理ブロック番号にシークします。	

S_STARTSTOP

● S_STARTSTOP　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄２Ｆ

機　能	ＳＣＳＩ装置に対して以降の操作を可能、または操作を不可能にすることを要求します。		
入　力	D0. L	$F5	ＩＯＣＳコール番号
	D1. L	$2F	ＳＣＳＩコール番号
	D3. L	操作可／不可	0操作不可 1操作可
	D4. L	ターゲットのＩＤ	
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了	
解　説	入力ののD3. Lが1のときはＳＣＳＩ装置を操作できるように設定しまた0のときはＳＣＳＩ装置を停止（操作不可）するように要求します。		

S_REASSIGN

● S_REASSIGN　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄３１

機　能	ＲＥＡＳＳＩＧＮ　ＢＬＯＣＫＳ		
入　力	D0. L	$F5	ＩＯＣＳコール番号
	D1. L	$31	ＳＣＳＩコール番号
	D3. L	書き込みバイト数	
	D4. L	ターゲットのＩＤ	
	A1. L	書き込み先頭アドレス	
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了	
解　説	欠陥ブロックの再割り当てを要求します。 A1. LからD3. Lのバイト数分のＲＥＡＳＳＩＧＮ　ＢＬＯＫＳデータを書き込みます。		

S_PAMEDIUM

● S_PAMEDIUM　ＩＯＣＳコール番号：＄Ｆ５　ＳＣＳＩコール番号：＄３２

機　能	メディアのイジェクトの禁止・許可を設定		
入　力	D0. L	$F5	ＩＯＣＳコール番号
	D1. L	$32	ＳＣＳＩコール番号
	D3. L	イジェクト許可／イジェクト禁止　0イジェクト許可 1イジェクト禁止	
	D4. L	ターゲットのＩＤ	
リターン	D0. L	ステータス（上位＝メッセージインフェーズで得た情報／下位＝ステータスインフェーズで得た情報） －１なら異常終了	
解　説	メディアのイジェクト禁止／許可を設定します。入力のD3. Lが0のときイジェクト許可、1のときイジェクト禁止に設定します。		

2.6 エラーコード

上位＝ＩＮＴＳ（ＳＰＣの割り込み原因）

	D23	D22	D21	D20	D19	D18	D17	D16		
INTS									0	0

- D16: ＲＥＳＥＴ　コンディション割り込み
- D17: ＳＰＣ　ハードウェア割り込み
- D18: セレクション　タイムアウト割り込み
- D19: 実行しようとする転送フェーズとＳＣＳＩ上で要求されている転送フェーズとが不一致、あるいは転送実行中に他の転送フェーズが要求されたときの割り込み
- D20: ＳＰＣに対するコマンドの動作の終了割り込み
- D21: ＤＩＳＣＯＮＮＥＣＴＥＤ　割り込み
- D22: ＲＥＳＥＬＥＣＴＥＤ　割り込み
- D23: ＳＥＬＥＣＴＥＤ　割り込み

各ビットは　0 ・・・・・・・・割り込み無し
　　　　　　1 ・・・・・・・・割り込み有り

下位＝ＰＳＮＳ（ＳＣＳＩバス上の制御信号の状態）

			D7	D6	D5	D4	D3	D2	D1	D0
PSNS	0	0	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O

各ビットは 0・・・・・・・・・・・信号ノンアクティブ
　　　　　 1・・・・・・・・・・・信号アクティブ

シャープ株式会社

本　　社　〒545 大阪市阿倍野区長池町22番22号
電子機器事業本部　〒329-21 栃木県矢板市早川町174番地
AVCシステム事業推進室
お問い合わせ先　〒162 東京都新宿区市谷八幡町8番地　電話 (03)3260-1161(大代表)
東京支社内　電子機器事業本部　AVCシステム事業推進室　ソフトウェア担当